TOWARDS A METHODOLOGY FOR COMPLEX ADAPTIVE INTERACTIVE ARCHITECTURE

PHD DISSERTATION
IR. TOMASZ JASKIEWICZ
PROMOTER: PROF. KAS OOSTERHUIS
CO-PROMOTER: DR. NIMISH BILORIA

Towards a Methodology for Complex Adaptive Interactive Architecture

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op 1 februari 2013 om 12:30 uur
door Tomasz Jan JAŚKIEWICZ
bouwkundig ingenieur
geboren te Gdynia, Polen.

Dit proefschrift is goedgekeurd door de promotor:
Prof. K. Oosterhuis
Copromotor Dr. Ir. N. Biloria

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. K. Oosterhuis,	Technische Universiteit Delft, promotor
Dr. Ir. N. Biloria,	Technische Universiteit Delft, copromotor
Prof. A. Saggio,	Sapienza University of Rome
Prof. D. Keyson,	Technische Universiteit Delft
Prof. P. Teuffel,	Technische Universiteit Eindhoven
Prof. A. van den Dobbelsteen,	Technische Universiteit Delft
Dr. W. Aprile,	Technische Universiteit Delft

ISBN 978-94-6186-109-2

Preface

Writing this dissertation has taken significantly more time than I might have originally expected. But it has also become a much broader, more diverse and more interdisciplinary endeavour than I might have ever anticipated. At the same time, however, I hope it is only a start of a longer journey. Upon finshing this dissertation I am deeply convinced that the time is ripe to put the outcome of this academic research to practice and attempt to realise the new kind of complex adaptive interactive buildings.

Not only the very idea of doing research in the nascent domain of interactive architecture, but also the broad spectrum and ambition of investigated here projects would never have been possible without the guidance and inspiration of my promoter prof. Kas Oosterhuis, my co-promoter dr. Nimish Biloria and the wonderful colleagues from Hyperbody and ONL[Oosterhuis_Lénárd]. It has also been invaluable for me to learn “the other ways of designing” from the colleagues from the ID-StudioLab who took me in to their lab after the disastrous fire of our old “Bouwkunde” and made me realise that listening to users may sometimes be more important than the designer’s ego or the drive to “play” with technology. Alongside my research I have received a tremendous amount of support and motivation from numerous inspiring people through discussions and debates we’ve had at TU Delft around the ideas of interactive architecture, participatory design, parametrics, cnc fabrication and so many other exciting topics in all possible ways connected to the present and future of architecture. I have to especially acknowledge the guidance I have received from dr. Axel Killian and dr. Walter Aprile who helped me find my way in the early period of my research work. I also have to thank Dieter and Chris, my stellar paranymphs, as well as Christian, Nora, Owen and Jelle for all the great work and fun we shared, mutual support and constructive critique that was always there when needed. I furthermore want to thank Stefan and Andrei for kick-starting Hive Systems and fuelling my hope that interactive architecture is now becoming reality. Last but not least, I would like to thank all the students who dared to enter taught by me courses, put their faith in me as their teacher and whose work provides a significant contribution to this dissertation.

Yet, all this would have never been possible without those closest to my heart. Thank you Ayreen for all your help and support throughout all the years of my PhD research, motivating me in the moments of doubt, helping me regain focus and always putting things back on the right track. Thank you my parents, grandparents and sister for supporting me in all ways possible at every step of my life and relentlessly trying your best to understand the uncommon topics of my work. All this, despite me being thirteen hundred kilometres away from you.

Contents

Introduction

The research presented in this dissertation stems from a multifaceted need of us, humans, to transform, improve and adapt our habitats. This need is confronted with inability of contemporary architecture to develop profoundly adaptable architectural living environments.

Author's experience prior to the outset of this research included work on utilisation of interactive technologies in experimental architectural structures and in architectural design processes. The initial hypothesis for this dissertation has been based on this experience. The hypothesis postulates that the degree and quality of architectural adaptation can be significantly improved by replacing traditionally centralised and hierarchical architectural systems by ones that are largely distributed, open and extensible.

Such postulated hypothesis implies creation of buildings consisting of very large numbers of adaptable building components. Such buildings are to operate as complex adaptive systems, making them capable of much higher degrees of adaptation, autonomy and robustness than any centralised architectural system could ever permit. Components in such architectural complex adaptive systems are to develop and maintain interactions with inhabitants of architectural systems and among each other. Those interactions are to function as main drivers of these systems' adaptation.

The extensive background research following the above intial hypothesis has shown that such *complex adaptive systems approach* is not new in architectural thinking and can be traced back to architectural writings and designs developed since 1950s until the present day. Nevertheless, in respect to creation of adaptive architectural systems, the theory has had little consequence on contemporary praxis. Complex adaptive building systems have not yet been developed beyond conceptual designs and experimental prototypes. At the same time, emergence of digital media, ubiquitous computing and the internet of things have triggered a recent revival of interest in adaptive and interactive buildings. Yet, to date, despite growing consensus on the validity of the complexity-driven approach towards creation of adaptive buildings, little systematised knowledge exists in this area.

Consequently, instead of developing yet another theoretical variation on the theme of complexity in architecture, the aim of this research has been focused on identifying practical problems that hinder present day development of complex adaptive interactive architecture and on gradually assembling a new framework for such architecture with the hope of providing grounds for methodologies for development of interactive architecture. The purpose of this work is to remove bottlenecks encountered in studied development of complex adaptive interactive architecture and to permit its further advancement. This task has shown to be highly intricate. As the result, the single focal point of the dissertation has delaminated into six parallel, yet tightly interwoven research trajectories, which are reflected in the six main chapters of the dissertation that follow the research framework put forward in the first chapter.

The first chapter presents a compact summary of background research, which will be further elaborated in chapters II and III. Based on this background, the framework of the research is concisely set forth, providing the canvas on which the dissertation unfolds in the later chapters. The purpose of this chapter is to provide a general structure in which further chapters can be situated and form a consistent, scientifically sound whole.

The second and third chapters provide an extended account of the performed background research from two distinct standpoints. The second chapter investigates the relationship between inhabitants and architectural habitats and subsequently attempts to scrutinise the need for spatial adaptation that stems from this relationship, while steering away from established architectural and cultural conventions and standards. The third chapter investigates interpretation of architecture as a complex adaptive system and the resulting ontological shift in perceiving and dealing with development of built spaces as complex adaptive systems.

The following three chapters trace design research experiments that were performed and studied throughout the entire research period. The aim of those experiments has been to directly assess applicability of investigated theories to practical architectural problems and to gradually formulate the new systematic and methodological approach towards development of complex adaptive interactive architecture. Those three main chapters have been organised based on the studied aspects of project development. Consequently, chapter four focuses on development and analysis of applied design methods. Chapter five investigates the role of new instruments developed to facilitate these processes. Eventually, chapter six deals with realisation and operation of studied experimental interactive architectural systems.

The seventh, chapter serves an integrating role. It extracts the findings from preceding it chapters and gradually assembles the building practice-oriented foundation for future processes of design, realisation and operation of complex adaptive interactive architecture. This foundation ultimately takes the form of an extensible project framework, which is evaluated in the concluding parts of the dissertation and provides grounds for the next generation of architectural projects yet to come.

I. Research framework

Summary:

The first chapter presents a compact overview of background research in the domain of interactive and adaptive architecture. Based on this investigation, the framework of the research is concisely set forth, providing the canvas on which the dissertation further unfolds in the following chapters.

1. Background

The term "interactive architecture" (iA) denotes an architecture capable of continuous self-adaptation to ever-changing conditions of its content and context. In other words, interactive architecture can be concisely defined as "architecture that exhibits autonomous behaviour, in which that behaviour evolves through interactions with its users and environment"[1]. Although no comprehensive examples of interactive architecture exist to date, development of interactive architecture is founded in a large body of theoretical work followed by numerous experimental projects. Demand for such architecture is assumed[2], but until comprehensive cases of iA are realised, this demand cannot be verified.

a) History of iA

Interactive architecture has been envisioned[3] since the 1950s[4]. It stems from the premise that buildings and built environments can be created in a continuous fashion in direct response to the actions of architecture's inhabitants. Early iA projects provided courageous visions of entire cities created through bottom-up interactions between buildings and their users. However, the disillusionment with technology that occurred in the 1970s has tamed the development of iA projects. The past decade has seen a revival of iA ideas[5], made possible due to technological advancements and following new societal trends (such as new media, do-it-yourself electronics and programming, lowering cost of computing, open source, smart materials, social networks, online communities, crowdsourcing, knowledge globalisation). Nevertheless, to date all cases in the domain of iA are experimental and highly limited in scale and scope.

b) Specificity of the iA worldview

The fundamental difficulty in dealing with interactive architecture stems from the fact that it is based on a fundamentally different worldview than traditional architecture[6]. This worldview is mainly characterised by the following traits:

- iA is a process as much as a product, in opposition to traditional architecture seen as a product only.
- Interaction implies indeterminacy of any iA processes, in opposition to traditional architecture being finite, designed and built in a top-down, fully predetermined manner.

[1] For further reference see section II.2.4

[2] Yet, due to lack of cultural reference and verified performance indicators, direct demand for iA does not exist.

[3] Studied under different names, for further reference see section II.4

[4] e.g. see projects of Nieuwenhuijs, Friedman, Price, Cook, Negroponte, Pask, Frazer, discussed in section II.3

[5] e.g., in works of Novak, Oosterhuis, Haque, Beesley, Fox and Kemp, d'Estree Sterk, Roosegaarde, discussed in section II.3.8

[6] For further reference see sections II.4, II.2.5

- Users are in the centre of the development and operation of any iA process, in opposition to traditional architecture where mostly designers, developers, engineers and stakeholders determine the spatial organisation and qualities of the built environment.

c) Relevance

Interactive architecture has potentially fundamental societal relevance. However, the exact societal impact of iA cannot be fully determined without further development of comprehensive built examples. The following social impacts are indicative and the verification of the assumed possibilities is not in the scope of this research. The following points are further elaborated on throughout chapters II and III:

- Creation of richer and more engaging experiences and spatial affordances in buildings than currently attainable.
- Creation of spaces closely fit to users' needs (optimization of response to direct demand).
- Optimization of spatial organization (integration of spatial affordances).
- Possibility of stronger proactive influence on users' activities, stimulation of users' needs[1].
- Sustainability, natural growth of buildings over time, optimization of energy use through active adaptation, continuous adaptation of buildings in place of rebuilding.
- Societal and cultural models of inhabitation to strengthen the participation of inhabitants in the building process and to augment/empower social interactions.
- Financial models for architecture based on direct participation in the transformation of inhabited space.

d) Interdisciplinary character

iA is a nascent domain and requires integration of research from various distinct research fields. For this reason, chapters II and III extend chapter I and provide further detailed overview and interrelation of background knowledge related to the iA domain, while integrating and reformulating definitions coming from these fields. The extensiveness of these two chapters goes beyond the typical format of a PhD dissertation. However, it is considered essential to expand the background research overview, since little comprehensive literature exists in the domain of iA that would comprehensively cover the state-of-the-art in the field.

Upon investigation of the technologies available for building automation and human computer interaction, it becomes clear that technological "ingredients" that are required for the vision of interactive architecture to be realised already exist. There is also a great demand for spatial adaptability grounded in numerous precedent examples and attempts of creation of adaptable architecture. The new paradigm of systems science that has found its way to most scientific disciplines and also offers ways in which complex problems encountered by interactive architectural systems could be dealt with. New, digitally driven, possibilities for virtual and physical creation of interactive architecture are already being broadly explored and many more lessons from systems engineering, computer science and interaction design can still be learned.

The radically new spatial qualities that dynamic buildings have potential to deliver require a new approach, unconstrained by past conventions and standards. Yet, there is still a notable lack of integration of all the above mentioned developments and no sound design methods nor frameworks exist that could further facilitate the development of interactive architecture. New, appropriate methodologies need to be defined and validated. Implications of creation and use of interactive architecture require thorough investigation and experimentation before applicable and reliable solutions can be brought to real-world applications.

[1] Ethical concerns are not in the scope of this thesis, need to be discussed per project

2. Problems of iA

It can be generally postulated that interactive architecture does not yet exist[1]. There are many buildings with actively changing components or installations. These include HVAC (heating, ventilation, air-conditioning), security, emergency, light and sound and others. Also sensor technologies and embedded computing is widely present in buildings and is used to gather information about building use, and internal and external occurrences and conditions. However this form of dynamic building alteration is typically automated and top-down controlled using Building Management Systems (BMS). Such form of centralised control is in opposition to the concept of interaction, which is inherently a bottom-up process. Existing reconfigurable buildings don't exhibit autonomous agency of their own required for interaction to take place.

a) Lack of comprehensive reference projects

Various aspects of interactive architecture have been preliminarily tested in experimental installations, but have not found their way to commercial applications. Without comprehensive case study projects, no evaluation of actual usability or performance of iA can be made.

b) Scalability

To date only limited in scale and scope interactive architectural installations are being developed. These explorations are typically seen as "art" rather than actual architectural or building engineering research. Large kinetically transformable or otherwise inherently dynamic building structures lack open interaction[2]. Centralised interactive systems are not scalable beyond a threshold of a number of actors in a typical pavilion-scale installation.

c) Society-embedded constraints

Legal, cultural, financial and technological constraints are among the main bottlenecks in commercial and large-scale development of iA[3] and require more comprehensive case study projects in order to be thoroughly investigated and overcome[4].

d) Lack of rules of conduct

New methods (including best practices, ontologies, techniques, instruments) are needed to design and develop more comprehensive iA case studies. These methods need to be holistically developed from ground up and be supported by novel design instruments and technologies[5].

3. Hypotheses

The hypothesis initially assumed for this research states that: "Degree and quality of architectural adaptation can be significantly improved by replacing traditionally centralised and hierarchical organisation of architectural systems by largely distributed, open and extensible one, leading to foundation of new methodologies for interactive architecture." This hypothesis can be extended by a consequent assumption that the proposed largely

[1] As further elaborated in section II.2

[2] As further discussed in sections II.3-4

[3] For further reference see sections II.1-4 and III. 1

[4] See section II.5

[5] For further detail see section III.3

distributed approach leads to formation of complex adaptive interactive architectural systems including large numbers of autonomous, adaptive agents. It is expected that under well-engineered conditions such processes will allow development of complex adaptive architectural systems with multifaceted benefits to broadly understood well-being of their inhabitants. The hypothesis is further grounded in chapters II and III. From these chapters alone it can be concluded that:

- Largely distributed approach to iA has been studied and consensus among experts exists that it is the preferred path for further advancement of iA.
- Further design research case study experiments can provide additional validation for the distributed approach to iA. However, the validity of the largely distributed approach cannot be fully proven without comprehensive iA case studies beyond the scope of this research and without a shared framework for iA projects.
- The lack of a shared development framework for complex adaptive interactive architecture blocks further application and validation of the largely distributed approach to iA.
- There is not enough data to assume any specific form of the shared development framework for iA and in this respect to postulate a specific framework as a hypothesis that could be proven or disproven in the dissertation.
- Consequently, the iA framework is to be seen as a theory formulated gradually throughout the executed design research case study experiments, following grounded theory research methodology and elements of the actor-network theory. The iA framework is thus to be gradually and rigorously constructed in the process of execution and evaluation of design research experiments.

4. Objectives

a) To further validate the largely distributed approach towards creation of interactive architecture.

The postulated approach can be theoretically proven to be the only scalable approach guaranteeing open-ended architectural adaptation (see chapter III. 2-4). However, no metrics or organisational models exist that would allow validation of the performance of this approach. The objective of this research is to lay foundations for such metric and framework, while providing initial evaluation of various techniques and methods on the path of realisation of complex adaptive interactive architecture.

b) To rigorously formulate an iA development framework allowing creation of comprehensive iA projects, providing the foundation for future iA methodologies and enabling the execution of the first research objective.

Provision of a comprehensive framework for iA development is the main objective of the research. It is aimed at facilitation of future iA projects, allowing them to reach higher levels of complexity and scale. Most importantly, however, it provides a point of reference for development of techniques, instruments and methods for realization of interactive architecture and consequent development and sharing of knowledge in this domain.

5. Research questions

a) What are the characteristics and features of the process of developing an interactive building as a dynamic complex adaptive interactive system?

As postulated in point 1., the process of iA's performance is inherently different from that of traditional architecture. This process has been discussed from the humanistic and theoretical side[1], however no concrete systematised analysis of the process of iA development exists. It is unknown what are the possible variations of this process, what are its ingredients and rules of conduct. This research aspires to provide foundations for answering this question.

b) What taxonomies and organisational rules are required for the development process of complex adaptive interactive architecture to unfold and sustain itself?

To date the development of complex adaptive architectural systems has been unstructured and fragmented. Created distributed iA installations have not been meant for usability beyond the confines of an exhibition piece or show (E.g. as in the case of ADA[2] or Hylozoic Ground[3]). When considering comprehensive development of iA in the practical realm, a more structured approach is required. Consequently, shared taxonomy and general rules of deployment are needed to both technologically and culturally bring iA to its real-life application.

c) What technological enablers are required in development of complex adaptive iA?

The autonomous operation of architecture and ability of architecture to interact, unless treated metaphorically, requires the use technology. Except for rare examples[4], digital sensing, data processing and digitally controlled actuation are required to enable such interactions. Additionally, the process of development and deployment of iA can be catalysed or largely facilitated by employment of various design instruments, novel materials and fabrication techniques. This research aims to critically evaluate what technological enablers, including among others facilities, instruments, techniques or materials, are needed to realise complex adaptive interactive architecture.

[1] Philip Beesley and Omar Khan, *Situated Technologies Pamphlets 4: Responsive Architecture, Performing Instruments* (New York: The Architectural League of New York, 2009).

[2] K. Eng et al., 'Ada - Intelligent Space: An Artificial Creature for the SwissExpo.02', in *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA '03*, vol. 3, 2003, 4154 – 4159 vol.3.

[3] Philip Beesley and Rachel Armstrong, 'Soil and Protoplasm: The Hylozoic Ground Project', *Architectural Design* 81, no. 2 (March 1, 2011): 78–89.

[4] Theo Janssen, 'Strandbeest', accessed July 29, 2012, http://www.strandbeest.com/.

6. Research methodology and strategy

6.1. Research methodology

In the context of the presented argument, too little knowledge exists to postulate any specific framework for integrated design and operation of iA, which could then be validated or invalidated through a set of devised design experiments, what would have been the most common way of conducting scientific research and applying the scientific method[1].

In social sciences, an alternative approach has been postulated under the name of the grounded theory. In grounded theory research, no theory is assumed as hypothesis at the outset of research. Instead, as research experiments are being conducted, "codes" are being extracted as patterns of collected data. Consequently concepts are defined, categories are set and eventually the theory is formulated *a posteriori* to the conducted research experiments.[2]

In case of grounded theory, the aim of theories developed in the research process is to develop an understanding of a phenomenon (often of social nature). In case of research on complex adaptive architectural systems, the analysed phenomenon, which is the creation and operation of complex adaptive systems, depends on the formulated theory. This adds a complication to the research methodology, but it does not contradict it. However, it requires an iterative approach when working versions of the theory are postulated along the research experiments and the application of these working versions is validated as the theory gains its shape.

In line with the general consideration for the grounded theory research method, actor-network theory provides the ontological foundation for the construction and navigation through the design research experiments. Actor network-theory has in itself been shown to be a valid design research methodology[3]. It provides thorough tools to analyse complex social phenomena and trace networks of dependencies, interactions and transformations through studied situations, without reducing them to constrained systems a priori to the conducted research. Tracing actor-networks can thus become a tool in which phenomena observed in design research can be mapped and consequently system "views" can be derived from such tracings as intentional reductions and generalisations of what was being traced.

For the definition of research experiments, design case study research has been chosen. As discussed by Richard Foqué, "Research by design tries to explore and change the world, and by doing so, tries to gain knowledge about how man analyses and explores the world and brings it into culture: how we create a man-made world. It does so by creating design applications, relying on technological knowledge and artistic interpretation"[4] Design research case studies are to be formulated in ways, that through their execution, a new insight is granted into the studied knowledge domain and new models can be constructed, contributing to the advancement of theory. Experimental, exploratory design case study research can thus be employed as source of qualitative and, to a lesser extent, quantitative research data and provides means for systematic validation of developed theory throughout the process of its formulation

[1] John W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 3rd ed. (Sage Publications, Inc, 2008).

[2] Kathy Charmaz, *Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis*, 1st ed. (Sage Publications Ltd, 2006).

[3] Ben J Kraal, 'Actor-network Inspired Design Research: Methodology And Reflections' Conference Paper, 2007.

[4] Richard Foque, *Building Knowledge in Architecture* (ASP - Academic & Scientific Publishers, 2010).

6.2. Research plan

The rationale of architectural adaptation and interactive architecture is not broadly acknowledged by the general public or by the majority of architectural community. The research on iA, including the worldview of seeing it as a complex system of things and people is scattered. Several publications attempting to integrate the novel field of iA exist (e.g. by Fox and Kemp[1], Oosterhuis[2], Bullivant[3]), however, they don't deal with the problem exhaustively. In order to steer further research in the domain of iA, a thorough argument for architectural adaptation needs to be formulated (chapters II-III) to serve as foundation for further research (chapters IV-VI). The development of an integrating framework requires case-study driven research into novel methods for iA development (leading to integrated design and operation). (chapter IV.) Findings of such research can lead to assembling the iA framework. The studied methods require support of new design instruments. Such instruments need to be developed alongside studied methods in order to validate these methods (chapter V). Operation of iA requires research into new ways of embedding technology in buildings in order to evaluate studied projects (chapter VI). The proposed framework can be constructed through the process of assembling the recursive conventions appearing throughout the case study projects (chapter VII).

In the context of presented problems, the framework for interactive architectural systems takes the role of the theory that is to be iteratively developed through a series of design case study experiments. The domain of research is the integrated design and creation of operational architectural systems. The operational architectural systems are set to be formed and studied as systems consisting of heterogeneous adaptive agents. These agents can be building components, humans and other living entities as well as non-embodied entities.

In consideration to this approach, the architectural system is acknowledged to be one of many possible views on constructed reality of architecture. The framework in which created systems operate is the subject of exploration and therefore it is open and extensible. The very nature of that framework is unknown and will be defined throughout the experiments.

Following the listed assumptions, research experiments will be conducted in three focus categories, namely: design methods, instruments and operation. The three categories will be cross-influenced. Throughout these experiments a hybridized framework for creation of interactive architecture will be developed and subsequently discussed in detail. The final research case study experiment will be an attempt to apply and evaluate the framework and will serve as foundation for critical discussion and an outlook into future.

[1] Michael Fox and Miles Kemp, *Interactive Architecture* (Princeton Architectural Press, 2009).

[2] Kas Oosterhuis, *Hyperbodies, Towards an E-motive Architecture*, ed. Antonino Saggio, 1st ed. (Birkhäuser Basel, 2003).

[3] Lucy Bullivant, *4dspace: Interactive Architecture*, 1st ed. (Academy Press, 2005).

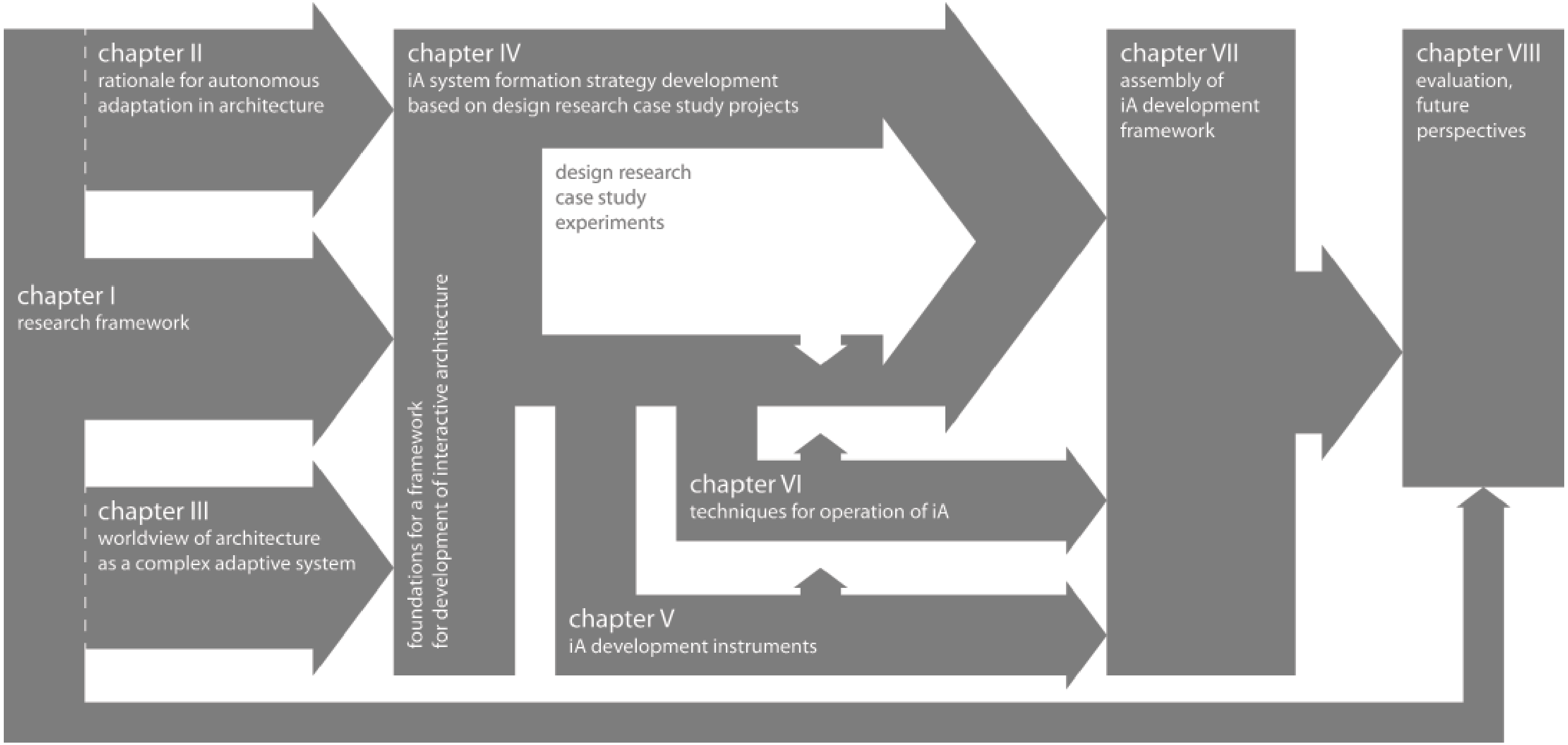
7. Boundary conditions and research context

The conducted research deals with a broad range of problems, while it is also executed within unavoidable constraints. The research is intended as a preliminary exploration in the nascent field of interactive architecture, and is expected to provide foundations for further more focused research projects, rather than to deliver definitive answers to all investigated problems.

Undertaken iA case projects is limited by timeframe, budget and organisational context of this research. For this reason, studied cases are constrained to the educational context or to projects executed within research embedded in ONL[Oosterhuis_Lénárd] through a structural research collaboration.

The research is interdisciplinary, and aside from architecture, building technology and aspects of urbanism, it also bares especially strong relationships to the domains of: interaction design, user experience design, multi-agent systems, embedded software and aspects of robotics.

The executed research is embedded in the research programme of the Hyperbody chair at the faculty of Architecture, the Delft University of Technology, including structural links to the development of protoSPACE laboratory and associated research initiatives. It is also embedded in the Hyperbody educational programme, including Hyperbody MSc design studio courses (various semesters set up, coordinated and taught by the author) and Interactive Environments Minor programme taught in cooperation with the ID-StudioLab at Industrial Design Engineering Faculty of the Delft University of Technology (co-initiated and taught by the author and set up in conjunction with this research).


chapter II
rationale for autonomous adaptation in architecture
chapter I
research framework
chapter III
worldview of architecture as a complex adaptive system
chapter IV
iA system formation strategy development based on design research case study projects
design research case study experiments
foundations for a framework for development of interactive architecture
chapter VI
techniques for operation of iA
chapter V
iA development instruments
chapter VII
assembly of iA development framework
chapter VIII
evaluation, future perspectives

Img. 1. Research framework overview diagram

II. Autonomous architectural adaptation

Summary:

This chapter comprehensively discusses the rationale of adaptation of buildings and other architectural structures, and subsequently introduces and investigates the prospects for autonomy of such adaptation. It does so in order to provide grounds for a detailed definition, raison d'être and clear direction for the development of "interactive architecture" (iA), establishing the point of departure for further research and foundation for the iA development framework.

At the outset, the broad phenomenon of architectural adaptation is examined. The given understanding of this phenomenon is based on the perspective of considering architecture to be both a product and a process of the adaptation of a human habitat to human needs. The changing nature of those needs and their bidirectional relationship with affordances of the human habitat is further discussed and contextualised (section 1.). This argument is subsequently followed by an organised overview of means, by which material adaptation of architecture is typically accomplished (section 2.). From here, the possibility of further enhancements of architecture's ability to adapt is discussed and a historical overview of concepts and projects relating to the idea of autonomous architectural adaptation is presented (section 3.). In consequence, the emergence of "interactive architecture" is discussed as a result of architectural adaptation increasing its speed and being performed autonomously (section 4.). Multiple aspects of societal relevance of interactive architecture are taken into account, based on conceptual scenarios and examples (section 5.). Key problems, risks, challenges and expectations towards interactive architecture are subsequently identified (section 6.).

1. Architectural adaptability

In the Western culture built spaces are traditionally perceived as fundamentally static. Two millennia ago, Vitruvius referred to solidity (*firmitas*) as one of the three main qualities of architecture[1]. In modern days, Louis Kahn praised architectural monumentality as a "quality inherent in a (architectural) structure which conveys the feeling of its eternity, that it cannot be added to or changed."[2] However, observing the development of any arbitrarily chosen human settlement over its history, we can notice that its architecture perpetually undergoes transformations on all its scales. Buildings are being worn out, renovated, remodelled, torn down and rebuilt, while their functions may frequently change in the process. In this way, architecture not only adapts to changes in its content and context but it can also be continuously improved and adjusted, answering to changing needs of its users and permanent transformations of its surrounding environment.

Reyner Banham's critique on traditional architectural profession and cultural perception of architecture holds true to this day: "Buildings were made to last (...). Architecture came to be seen as the conscious art of creating conscious massive and perdurable structures, and came to see itself professionally as no more than that art, which is one of the reasons for their present problems and uncertainties. Societies (...) prescribe the creation of fit environments for human activities; the architectural profession responds, reflexively, by proposing enclosed

[1] Vitruvius, *Vitruvius: Ten Books on Architecture* (Cambridge University Press, 2001).

[2] Louis Kahn, *Louis Kahn: Essential Texts* (W. W. Norton & Company, 2003).

spaces framed by massive structures, because that is what architects have been taught to do, and what society has been taught to expect from architects. But such structures may be open to objection on the number of grounds"[1]. It is implied in such critique that architects, and ultimately the entire society, should, rather than perceiving architecture through its fixed forms, understand it through ways in which it is used and through ways in which it operates in its environment. Be this environment the set of natural, cultural, or any other forces having a continuous affect on architecture. In such altered worldview, architecture can be seen, as postulated by Henri Lefebvre, as space, which is continuously "produced" by social and cultural everyday activities of its inhabitants.[2]

Such critique calls for a new approach to architecture, going beyond the idea of buildings being just permanent objects made of "stiff" matter. It calls for looking past their fixed, prescribed functions and past their presumed cultural meaning. It also calls for understanding how built spaces perform over time, in a complex relation to external and internal environments in which they are set to operate. For this, a broader look at architecture is required. It becomes essential to see architecture through the prism of its most basic role – spatial formation of human habitats.

The human habitats of today greatly differ from the natural ones. Human civilizations grew alongside humankind's continuous strive to free itself from the hindrance of harsh, changing and often unpredictable traits of the natural environment. (Shelters provided independence from weather and climate. Through farming, breeding of animals, processing, conserving and efficient distribution of food humans gained independence from the natural food sources. Transportation and communication technology diminished the significance of physical distances between places. Development of media allowed storing, transmitting and sharing of knowledge without the need of direct personal contact. Artificial light allowed partial independence from the natural day-night cycle.) Consequently, in the context of all kinds of adaptations performed by humans to their habitats, architecture can be defined as a specific sort of such adaptations that deals with the organisation of space. In other words, architecture can be broadly defined as *adaptation of space to human needs.*

Translation of basic human needs to architectural demands is a complex process. Needs of an individual are closely entangled with his or her beliefs, which together influence the daily activities that he or she may perform, often in conjunction with activities of other individuals. Architecture is created in order to facilitate those activities.

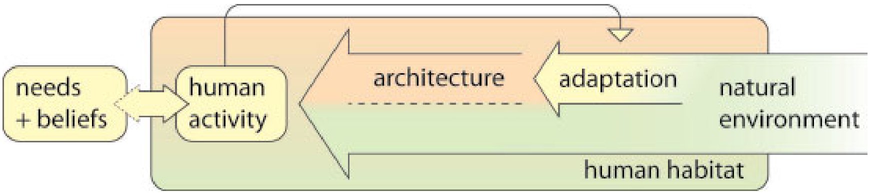


Fig.1. Reciprocal relationship between human activities and architecture seen as adaptation of the natural environment.

In order to fully understand this process and to devise a comprehensive model of architectural adaptation, elementary human needs must first be defined. Subsequently, mechanisms governing the relationship between human needs and human activities can be analysed. Eventually, different aspects of adaptation of the human habitat to those activities and the reciprocal influence of those adaptations on human needs and beliefs can be discussed.

[1] Reyner Banham, *The Architecture of the Well-Tempered Environment*, 2nd ed (Chicago: University of Chicago Press, 1984).

[2] Henri Lefebvre, *The Production of Space* (Wiley-Blackwell, 1992).

1.1. Adaptation of space to human needs

The most recognised classification of human needs has been formulated by Abraham Maslow. According to Maslow, human needs can be hierarchically organised in five categories, namely: physiological, safety, love and belonging, esteem and self-actualization[1], where satisfaction of lower, more fundamental, needs triggers the higher ones. However, Maslow's hierarchy can be criticised on grounds of human needs having been observed not to be strictly hierarchical. Contrary to Maslow's theory, human needs also exhibit a degree of accomplishment rather than a binary accomplishment condition[2] (e.g. being slightly hungry does not fully accomplish the need for food, but it does not have to stop an individual from seeking esteem of others). As an alternative, Kenrick et al. propose a hierarchy of consequent, yet less inter-dependent needs, built upon the order of evolution of needs in human species, rejecting the self-actualization needs altogether[3]. Among other models, the one of Alderfer[4] proposes three main categories of needs, namely: sustainment of one's own existence, relatedness to others and individual growth. In Alderfer's model, needs from those three categories may occur simultaneously to each other. Satisfaction of needs in one category strengthens needs in the higher one. Frustration occurs when needs are not satisfied, which leads to strengthening of needs in lower categories. Due to its openness, flexibility of interpretation and empirical accuracy, Alderfer's model provides a good foundation for further argumentation and positioning of human needs in the context of architectural adaptation.

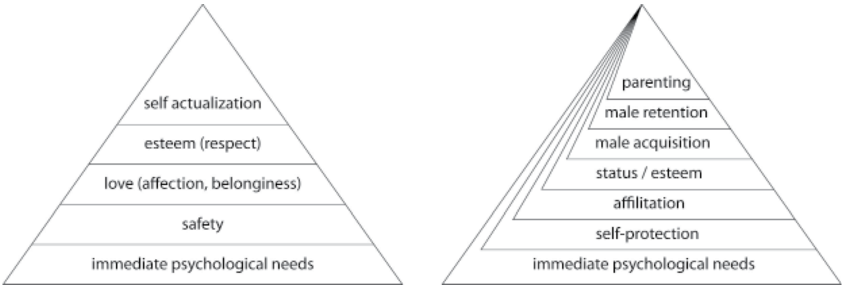


Fig.2. Maslow's hierarchy of needs, Kenrick's pyramid of needs

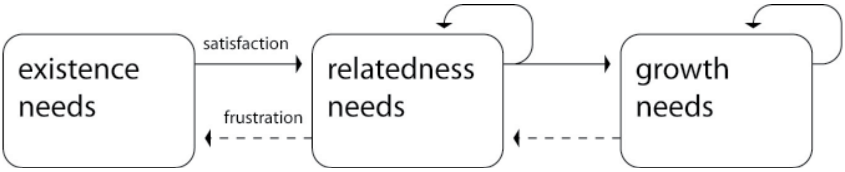


Fig.3. Alderfer's model of needs

In all models of human needs, human needs differ among individuals and change over time. These changes can by cyclical, following daily, weekly, monthly, yearly, and lifetime cycles, but are also highly dependent on individual's non-recurring biological conditions. Interpretation of human needs is also strongly affected by individual's "beliefs". In this context, beliefs can be broadly defined to comprise of cultural influences (including religion) and continuously acquired and revised (subjective) knowledge based on past experiences. Because of this,

[1] A.H. Maslov, 'A Theory of Human Motivation', *Psychological Review* 50 (1943): 370–396.

[2] Mahmoud A. Wahba and Lawrence G. Bridwell, 'Maslow Reconsidered: A Review of Research on the Need Hierarchy Theory', *Organizational Behavior and Human Performance* 15, no. 2 (April 1976): 212–240.

[3] Douglas T. Kenrick et al., 'Renovating the Pyramid of Needs', *Perspectives on Psychological Science* 5, no. 3 (May 1, 2010): 292 –314.

[4] Clayton P. Alderfer, *Existence, Relatedness, and Growth; Human Needs in Organizational Settings* (Free Pr, 1972).

beliefs are highly subjective and can greatly vary not only among cultural groups, but even among closely related individuals. There is a mutual interdependence between human needs and beliefs. Strong beliefs may lead to suppression or stimulation of specific needs. Together, needs and beliefs are the main drivers of human behaviour and govern ways of human's responses to external factors.

Based on a simplified BDI model of human reasoning[1], needs of an individual in conjunction with his or her beliefs produce desires. Those desires lead to intentions of actions and eventually to actions themselves. Aggregations of actions, possibly involving many individuals, can be referred to as activities. Consequently, human activities can be considered as expressions of needs and beliefs of individuals or groups. In turn, enactment of activities provides feedback to subsequent evaluation and alteration of needs and beliefs, being the foundation of the process of learning, which alters patterns of occurring desires, intentions and actions.

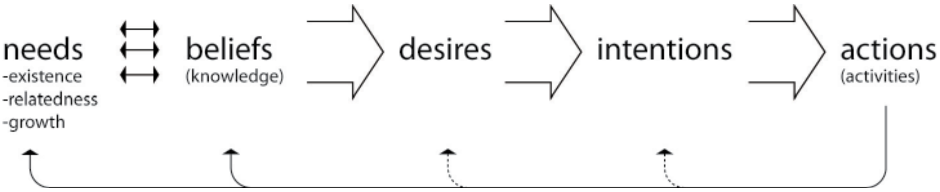


Fig.4. BDI model of human reasoning combined with Alderfer's model of needs.

Although human activities change over time and are a result of many fuzzy factors, they do have a tendency of forming recurring patterns. Many of such patterns follow daily and yearly cycles of nature, while they may also be strongly conditioned by culture. Variations in performed activities demand different affordances from the environment. As nature only provides rudimentary affordances, humans have learned do adapt their environment to increase its affordance to the most commonly occurring activities, where architecture in its broad sense is the entirety of the artificially adapted human habitat.

The most basic role of architecture can be considered to be the provision of inhabitable spaces that cater to human "existence" needs. These needs are individual, but also generic and easy to anticipate. (In modern days this not only means protection from the atmospheric conditions, but also access to running water and electricity, sanitary, food storage and preparation facilities, provision of fresh air, sustainment of comfortable heat and humidity levels, as well as means to regulate access to the inhabited space by its users. It is also critical that the inhabited space provides infrastructure to acquire food and perform its own maintenance, thus in spatial terms it can be translated to accessibility of e.g. shops and workplaces.)

The needs of "relatedness" are enforced by satisfaction of "existence" needs. Architecture may provide vast spatial means to enforce social connectivity, group belonging, as well as self- and mutual esteem of their inhabitants (which traits are often reflected in employment of architecture to mark social status). The architectural facilitation of "relatedness" needs involves spaces for collective use, allowing development and nurture of various kinds of relationships between people.

Eventually, architecture can also cater to the "growth" needs of its inhabitants. "Growth" needs are highly personal and facilitating them requires individual customisation.

Understanding human needs can provide a general idea of what spatial features may be required by people in given circumstances. However, ways in which needs are translated to specific human activities (and thus what specific architectural features they require) highly depend on people's individual "beliefs". "Beliefs" represent subjective knowledge of

[1] Michael E. Bratman, *Intention, Plans, and Practical Reason* (Center for the Study of Language and Inf, 1999).

an individual and involve lasting effects of any past experiences, acquired skills, cultural influences or religion. The role of “Beliefs” in presented model is interpretation of needs and formation of specific desires and intentions leading to actions and activities. (Religious factors may determine the kinds of food an individual would refuse to eat. Psychological phobias acquired through traumatic past experiences may result in refusal in commitment to otherwise “needed” activities. Cultural differences may prohibit the connectivity between individuals. One’s moral principles may set one’s direction of personal growth, etc.).

Following the model of human activity driven by needs and beliefs, adaptive performance of architecture can be analysed from two perspectives. Firstly, the focus can be placed on the changing natural and artificial environment and adaptation-of this environment to generic human activity patterns. Secondly, the focus can be placed on the perpetually changing activities of inhabitants driven by their fluctuating needs and beliefs and the habitat’s continuous adaptation-to those activities. The eventual model of architectural adaptation is a convergence of these two perspectives, where both the environment and the activities of people it adapts to are continuously in motion.

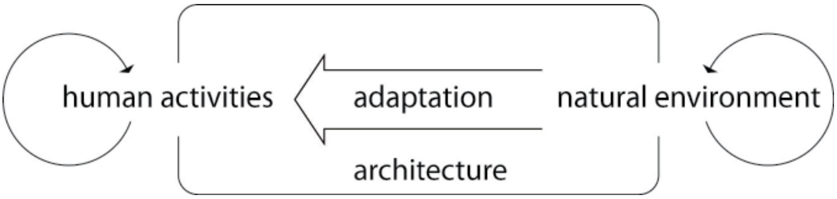


Fig.5. Architecture as a product of adaptation of the changing natural environment and adaptation to changing human activities.

Adaptation of space to a particular pattern of human activities (driven by human needs and beliefs) results in the concept of architectural function. Architectural function is a general term, which, in the context of discussion presented to this point, can be defined as a comprehensive set of affordances of an architectural space allowing a pattern of activities to be performed in this space in order to serve a certain purpose. Such purpose can be very specific (e.g. in a power plant function, the purpose is production of electric energy, which requires very specific architectural conditions), or inexplicit (e.g. a park function where the purpose is recreation of inhabitants, which can be achieved through a great variety of architectural means).

The primal function of architecture is shelter; protection of its inhabitants from the harmful conditions of the natural environment. Along with the development of human civilisation, the number of architectural functions has greatly proliferated. Following the popular design guidebook, main groups of functions are: residential, retail, office, public buildings, public urban spaces, roads and streets, education and research, workshops and industrial, agricultural, transportation, hotels, restaurants and cafes, zoos, aquariums and amusement parks, theatres and cinemas, sport and recreation, healthcare, places of worship, cemeteries and crematoriums and specialised buildings[1]. However, regardless of seeming comprehensiveness of such list, any classification of architectural functions is bound to remain incomplete due to the exponentially greater number of activities these functions cater to and endless possible combinations of those activities resulting in uncountable possible functional patterns. Yet, in all cases, provision of an architectural function involves transformation of the original natural environment, altering its conditions, its spatial organisation and providing required facilities.

[1] Ernst Neufert et al., *Architects’ Data*, 3rd ed. (Wiley-Blackwell, 2002).

1.2. Adaptation-of the natural

Conditions of the natural environment are not static. The natural environment undergoes continuous transformations. Many of those transformations (such as most geological processes) occur slowly enough to be neglectable from the perspective of one person's lifetime. Other conditions of the natural environment (such as weather) change very frequently. The natural environment also comprises of living organisms, ranging from bacteria and microorganisms, through plants, to large animals. Presence and activities of living organisms is essential to sustain our habitat. It also greatly contributes to the qualities and dynamic nature of the natural environment.

Many changes that occur in the natural environment are cyclical, directly or indirectly resulting from the movement of earth around the sun and around its own axis, being additionally augmented by the circulation of the Moon around the Earth. Most other changes occurring in the natural environment happen in consequence of those cycles and are often subject to similarly reoccurring patterns themselves. However, as chains of influence between natural occurrences aggregate, many occurences happening in the natural environment become highly unpredictable.

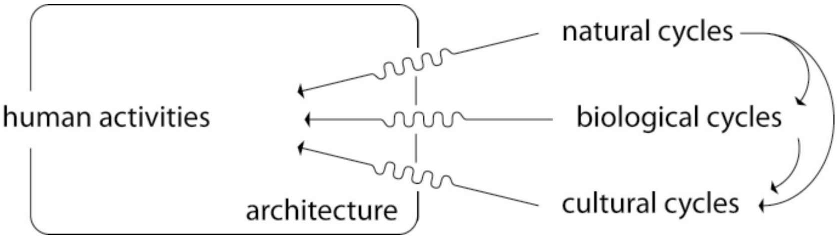


Fig.6. Affect of natural, biological and cultural cycles on each other and on human activities - modulated by architecture

Consequently, architecture can be considered to be an adaptation of a dynamic natural environment system, which exhibits qualities of various temporal characteristics, that are (semi)permanent, or undergo cyclical or non-recurring changes. Architecture can selectively either counteract the changes of the natural environment (e.g. using artificial light at night), modulate them (use of window blinds on a sunny day), or undergo change together with nature (e.g. a park).

a) Day-night cycle

The sun is the main source of energy on earth and the day-night cycle defines the main rhythm of life. Daytime provides sunlight and thus energy for plants to grow and heat required by most living organisms. Nights are colder and dark, when most organisms begin their rest, while many predators take advantage of night conditions and hunt for their prey. Periods of day and night vary across the globe and change cyclically throughout the year, with most extreme cases being the polar circles, where during summer sun never sets and during winter never rises. Although artificial lighting has allowed architecture to become independent of the day-night cycle, it still governs the main rhythm of human lives.

b) Yearly cycle

Weather is an important factor governing the activities of people. Weather conditions consist of factors such as temperature, amount of direct sunlight, rain or snow, wind, humidity and air pressure. Accuracy of anticipation of weather conditions is currently impossible for a period longer than one week. However, a general probability of occurrence of specific weather conditions varies according to the time of the year. Architectural adaptation to yearly cycle involves heating and air-conditioning systems, providing continuous heat comfort throughout the year. Artificial lighting helps to reduce the significance of changing day length throughout the year. Many previously seasonal activities can be now performed year-round.

c) Human life cycle

They life cycle of a human being dictates the types of activities a person is likely to engage in. The early part of life is spent on education, the middle part on work career, finding a partner and raising children and eventually the old age leads to retirement and decreased amount of activity. The average time in which a person becomes independent from parents varies across cultures and is still often associated with marriage and starting of own family. As children are born living environment requires more space. The eventual moving out of children reduces again the need for space. Retirement further changes daily activity patterns, while also increasing the need of external care. Because of these factors related to the life cycle an individual may change the place of life numerous times. Countless other factors, often related to work or culture may also stimulate change of living location throughout lifetimes of its users.

d) Plant and animal lifecycles

Similarly to humans, plants and animals have their own lifecycles, with length spanning from hours (insects, e.g. mayflies) to thousands of years (some species of plants and animals e.g. Antarctic sponge or Great Basin Bristlecone Pine). The complexity of the food chain on earth make relation between climate related cycles and lifecycles of living organisms highly interdependent. The disruption of those dependencies in result of human activity poses a threat to sustainment of human habitat on the global scale.

e) Non-cyclical changes of the environment

Although regularly recurring cycles have high influence on the natural environment, much of the occurrences in the natural environment are difficult to predict and have a non-recurring nature. This is due to a high number of interrelated factors at play, influencing independently every material particle and every living organism. Eventually, starting from the continuous drift of continents, through evolution of all living species, changing sea current patterns, climate, behavioural patterns of animals and people, down to continuously changing weather and natural cataclysms- the entire natural environment perpetually changes and transforms itself in a non-predetermined manner. Among non-cyclical changes of the environment are natural disasters; natural occurrences which go beyond the threshold of architectural ability to adapt its internal conditions to, resulting in destruction of buildings and injuries or loss of human lives.

1.3. Adaptation-to the individual and to the collective

Persistent transformation of the natural environment is one factor requiring architecture to continuously re-adapt itself. On the other hand, the main driver of architectural adaptation; the activities of people, also continuously change.

Many human activity patterns directly follow the natural cycles, while others follow independent rhythms. Yet, many human activities also change and evolve over time in a non-recurring manner. These changes may be entirely individual, but can also occur collectively throughout cultures.

a) Cyclic activity patterns

Many human activities synchronously follow the natural cycles. The cycle of day and night defines the primary rhythm of our lives, organising periods of sleep and rest and periods of activity. Types of human activities vary therefore based on the time of day or night. Daytime is reserved for work and is typically spent away from home. Evenings are reserved for social and relaxing activities, while most of the night is used for sleep. Following this rhythm, every morning, offices and factories fill up with people, to be abandoned in the evenings, when restaurants, cinemas, shopping malls and clubs begin to thrive. At night human habitats become quiet with most of the population asleep. Modern technologies enforced the specialisation of places for work, living, entertainment and other specific functionality, leading to frequently radical partition of places hosting those functions. Cities such as Los Angeles, where this separation is particularly evident provide vivid examples, with regular traffic congestion in rush hours manifesting the magnitude of the daily activity cycles shared by large human masses. Specific conditions can allow people to re-combine places for rest, work and leisure. Internet makes it possible for many independent knowledge workers to work from home; media technologies bring entertainment to house premises. On the far end of the spectrum, many closed communities, ranging from kibbutz's to arctic research stations, out of choice or necessity provide all daily life functionality within the premise of one, tight settlement. In many cases the change can be temporal. Going on holiday means moving to the place where living and entertainment are possible in one location and where work is not necessary. Students moving to a dorm room at a university campus choose integration of living, work and leisure in one location for the duration of their studies allowing for tighter integration of those activities.

Architecture provides means for adapting the environment to people's everyday cycles. Creation of places for rest, work and social activities is driven by how this cycle is organised and can differ greatly not only between cultures or sub-cultures but also between individuals of the same background. Because of the daily cycle, the occupancy of buildings throughout the day is not continuous and varies throughout the day and night. However, cyclical variations occur also at larger timescales.

Although a 7-day week is not directly related to any cyclical pattern occurring in nature, a 7-day period has culturally become a worldwide standard length for recurring patterns of cultural behaviour (although shorter and longer "weeks" existed locally during short periods of human history. E.g. 10 days in ancient Egypt and in France after the French Revolutions, 5 and 6 days in Soviet Russia, 13 and 20 days in Mayan calendar). In most cultures Saturday and Sunday form a weekend and are dedicated to rest, leisure and private duties. In some Muslim cultures Friday and Saturday form the weekend. Overall, the weekends disrupt the workday cycle introducing more diverse patterns of human activity on those days, which vary not only on cultural, but also on individual basis.

Beyond the weekly cycle, traditionally, the yearly cycle dictated types of work activities for most of the human population. In Europe, the mostly sunny and long spring and summer days were dedicated to work and accumulating supplies, while short days of the fall and winter, when food was scarce and weather conditions harsh were the time of stagnation, living off the previously gathered supplies. In areas closer to the equator the differences between seasons may be less radical, manifested by wet and dry seasons or changing monsoons and almost disappearing in equatorial rainforests where weather patterns don't significantly change throughout the year.

Technology in pair with architecture allowed people to become highly independent of the yearly cycle. We can create artificial light and warmth inside buildings during long nights of cold winters and cool down the living spaces during warm summer periods. In this way the typical daily and weekly cycle can continue throughout the year almost unchanged, yet in countries far from the equator social and cultural activities that involve use of outdoor locations cause large differences between types of activities performed by people in different seasons. Maintenance of artificial climate conditions also requires substantial use of energy, highly contributing to the scarcity of non-renewable energy sources.

b) Fluctuation and evolution of activity patterns

Patterns of activity change individually for each person throughout his or her lifetime. Long term changes are related to age. However, many short term changes can also be observed among individuals. Especially during one's adolescence and early adulthood these changes are most significant, when individual preferences are subject to on-going variation, which is reflected e.g. in patterns of physical activity[1]. In general, individual's activity patterns can be considered to evolve over his or her lifetime. If cultural, social, biological or environmental conditions of one's life change, his or her activity patterns adapt accordingly. A flux in activity patterns can also be observed on daily basis, based on singular day-to-day experiences. Many changes in activity patterns also occur as a result of cultural and technological advancements.

Although activity patterns differ among individuals, there is strong correlation between them, as many of such changes are caused by same or similar factors. Non-cyclical changes in activity patterns can be observed as general trends. Many trends, such as fashion are subject to fluctuations, while many other trends are more permanent. Non-recurring changes can occur collectively throughout cultures or sub-cultures. Generally speaking, the notion of "culture" can be defined as a shared pattern of knowledge, beliefs, behaviours and values. Similarly to genes providing the foundation to biological evolution, "memes", as defined by Dawkins[2] are cultural units to which alike evolutionary processes apply. Thus, analogically to the natural environment, human culture is continuously evolving. Culture of human groups has never been a constant factor, yet changes in culture have historically occurred slowly as cultural patterns were passed on between generations and gradually transformed in that process (with exception was revolutions and other types of turmoil, when cultural changes were occurring more radically).

Driven by technological advancements and resulting free flow of people and information, human cultures evolve at ever increasing pace. Contemporarily, patterns of human behaviour are decreasingly governed by historically established rhythms and more frequently new lifestyle patterns emerge spontaneously among groups that may have differing cultural patterns and backgrounds. This results in increasingly non-uniformity of ways in which lives of people are led. New technologies fuel creative development of new cultural patterns

1.4. Adaptation-of activities

Architectural adaptation of the environment in order to make it fit for activities of its inhabitants may appear as a single-directional process. In such process both the environment and the activities of its inhabitants are variable. Environmental conditions affect activities of people. This means both; conditions of the natural environment, as well as conditions artificially created though adaptations of the natural environment.

In this process, conditions of the environment are adapted to activities of people, but simultaneously, through a feedback loop, human activities are determined by the spatial conditions of human habitat. The influence of environment on human activities may have several forms.

[1] C J Caspersen, M A Pereira, and K M Curran, 'Changes in Physical Activity Patterns in the United States, by Sex and Cross-sectional Age', *Medicine and Science in Sports and Exercise* 32, no. 9 (September 2000): 1601–1609.

[2] Richard Dawkins, *The Selfish Gene*, 2nd ed. (Oxford University Press, USA, 1990).

a) Affordance

An affordance is a quality of an environment (or object), which allows an individual to perform a specific action. Clearly, individual's choice between different actions to perform is influenced by the fact if those actions can be performed in the given environments and, which is easiest to perform in the given circumstances. Thus particular architectural affordance influences activities and through reoccurrence leads to establishment of altered activity patterns.

Architectural affordance is conventionally attributed to architectural functions, where a "function" is a notion broader than "activity", burdened with a strong intention of predefined pattern of use.

b) Intention

Architecture is typically created with an intention to accommodate a specific kind of activities. Through provided affordances, or through other properties, such as culturally established patterns of spatial organisation, attributed symbols or other characteristics, architecture provides information about the intention of its use, which can further be culturally or legally enforced. (e.g. with a simple aid of an information sign, space with otherwise obvious affordance of a picnic ground becomes a quiet memorial or cemetery. n established and legally enforced pattern of use of a highway discourages from holding there a spontaneous game of soccer.)

c) Affect

Affect means an influence that an environment may induce on one's psychological or emotional state. Stimulus of light, sound, smell, consumed substances or specific cultural conditions can psychologically affect users, influencing their desires, intentions and actions.

1.5. Adaptation-to anticipated changes

Taking into account the possibility of patterns of inhabitant activities becoming altered in effect of a spatial intervention, those adaptations can also be performed pro-actively in response to an anticipated demand. Henry Ford is believed to have said "If I'd asked people what they wanted, they would have said a faster horse". This shows that although certain functionalities may not be directly needed by their potential users, a provision of those functionalities can change generate new needs and patterns of behaviour and activity. Architecture can operate similarly. Many buildings are created by developer firms in anticipation of a demand for specific functions, whether it's suburban housing or a shopping mall. Affordable suburban housing may entice a family used to an urban lifestyle to move to a suburban area and change their entire activity pattern accordingly. Proximity of a shopping mall can alter shopping and leisure habits of entire communities.

1.6. Re-adaptation-of the artificial

Since both the natural environment and the activities of humans taking place in their habitat continuously change in cyclical and non-recurring ways, often reciprocally influencing each other, and since adaptation of the natural environment to afford activities of its users can generate new activity patterns and alter the natural environment, it is unavoidable that the process of adapting the human environment needs to be continuous. The following section delves deeper into different aspects of re-adaptation of existing architecture.

2. Building for change

There are innumerable examples of architectural structures re-adapting to cyclical changes and fluctuations of user demands and of factors of their environment. There are many ways to classify such architectural transformations. Most typically found classifications are segregated based on function. Common function-related categories include for example “domotics”, “industrial automation”, “monument renovation” or “stage design”. In case of such classifications it is assumed that the function remains constant throughout the transformation process. Another way of classifying adaptive buildings is by the technical processes in which their transformation is achieved. For example Zuk distinguishes categories of: “kinetically controlled static structures”, “dynamically self-erecting structures”, “kinetic components”, “reversible architecture”, “incremental architecture”, “deformable architecture”, “mobile architecture” and “disposable architecture”[1]. In many cases the areas of concern are even further narrowed down by combining the functional specificity with one technical approach. In this way specialised types such as “portable homes” can be defined.

Gijsbers provides a different approach to systematization of building adaptation processes, presented in conjunction to values (vitality, functionality, esthetics, socio-cultural, ecological) and resulting demands and requirements. This systematization is further followed by analysis of adaptive processes in buildings, separated into flexibility in the design process and flexibility in the use process[2].

The classifications presented above provide systematic ways to study existing processes, however, can also induce limitations on formulation of new approaches. For example, Gijsbers’ approach makes it impossible to propose flexible processes where design and use of the building are both continuous and intertwined. For this reason, such classifications will be further avoided in this dissertation, since they discourage combinations of techniques and trans-functional adaptations. In order to provide a general overview of all kinds of adaptive transformations occurring in architecture with the intention of finding novel solutions, a different approach needs to be taken.

Kronenburg categorises “flexible” architectural structures based on their four mutually non-excluding characteristics: adaptation, transformation, movability and interaction[3]. It is the degree in which a given project satisfies one of these characteristics more than other ones that places it in a given category. Yet, this classification is still not satisfactory, as ultimately many projects could equally satisfy all characteristics. In order to formulate classification more adequate for the scope of research to follow, examples of adaptive architectural projects have been categorised based on the performance of the process of their adaptation. This process has been characterised by two variables leading to formation of four loosely defined groups of projects.

The first considered variable is the frequency of occurring spatial transformation. It is of significant importance whether a building adapts only several times in its lifecycle, or whether its adaptations occur more often, sometimes within minutes or seconds from one another.

The second chosen variable is an ability of created structures to adapt to initially unforeseen circumstances. *Unforeseen circumstances* are here defined as all possible internal and external factors that have influence on the designed space, but have not been accounted for and/or not existed when the space was originally designed and built. Such unforeseen circumstances may be of various sorts. External factors may be environmental “disasters”

[1] William and Roger H Clark Zuk, *Kinetic Architecture.* (Von Nostrand Reinhold Company., 1970).

[2] Roel Gijsbers, *Aanpasbaarheid Van De Draagstructuur : Veranderbaarheid Van De Drager Op Basis Van Gebruikerseisen in Het Kader Van Slimbouwen* (Eindhoven: Bouwstenen Publicatiebureau, 2011).

[3] Robert Kronenburg, *Flexible: Architecture That Responds to Change* (Laurence King Publishers, 2007).

such as hurricanes, earthquakes, floods, but also more persistent ones such as climate changes. Changes in average temperature, humidity or sun exposure are few of many examples of external factors that may have a significant influence on many features of buildings and their spaces. External factors may also be derivatives of various human actions such as changes in infrastructure and other interventions in areas surrounding the project. Many more factors of cultural, social and technological origin have to be also regarded as unforeseen circumstances that may have an even greater influence on the project. They affect architectural spaces not only externally, but mainly through building users that operate within the project itself. Most commonly, such changes result in functional demand alterations. To give the most basic examples, appearance of computer technology in offices radically changed office organization, requiring also additional spaces within floors, walls or ceilings for network infrastructure. Similarly, resulting cultural changes such as this caused by introduction of television (an average contemporary American spends 142 hours a week in front of a TV screen) required radical changes in spatial organization of a typical household. Such changes may happen gradually, but in the contemporary society their frequency and impact on built spaces keeps exponentially increasing. This variable can only be measured relatively and speculatively.

The third factor of relevance is the scale of transformation. Different kinds of transformation can occur simultaneously at different scales. However, the following classification has been deliberately performed with disregard to the scale of analysed transformation. A separate section focuses on the relationship between outlined categories and scale of their occurence.

In this way, four distinguishable categories of building adaptation have been defined, namely; reconfigurable, portable, flexible/adjustable and automated architecture. These groups are not mutually exclusive and approaches they represent can be in many ways combined.

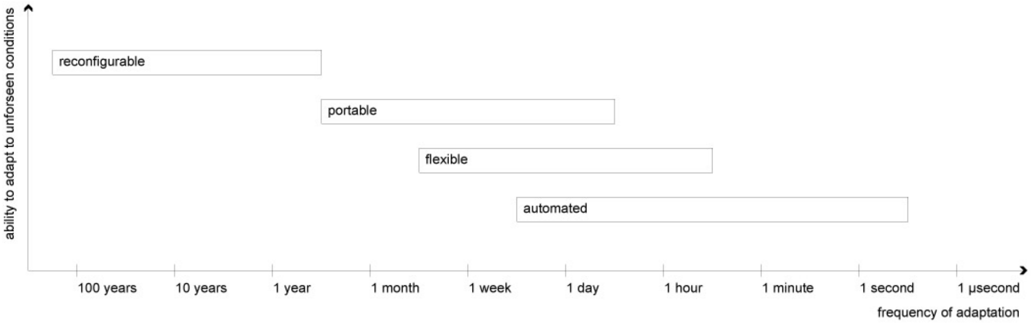


Fig.7. Four distinguished types of architectural adaptability

2.1. Adaptation by reconfiguration

“Building reconfiguration” can be broadly defined as transformation of building’s spatial organization. This can be normally achieved by disassembling selected building parts and assembling them in a different way as well as adding and/or removing building elements to the assembly. It usually involves temporary, complete or partial dysfunction of a building during the duration of reconfiguration process, typically performed by specialized workforce, not building users themselves.

Primitive building types, like those that still can be found in many slum or favela neighbourhoods, possess high reconfiguration capacity. Their spaces directly enable various activity routines and cater to needs their inhabitants. Thus if an additional room or facility is needed, due to simple building techniques and low standards for used materials new building features can be instantly added to a previously existing part of the structure, providing that external factors allow for such modification.

Along with technological advancements, such transformations became less common and ultimately impossible to perform by inhabitants alone without specialized assistance. Logical conclusion can safely be drawn that the cost and time needed to reconfigure the building increases with building's complexity, both in respect to its size and technology. Nevertheless, numerous examples of "parasitic" extensions to existing buildings and adaptations of unused old buildings for new functions show that buildings are always to a certain degree open for adaptive modifications.

"Reconfiguration" of buildings has been attempted to be re-facilitated along with the introduction of modular building components. Among some of the earliest realized projects on large scale of this kind is built in 1972 the Nagakin Capsule Tower by Kisho Kurokava. Although it was reconfigurable in its design, such reconfiguration never took place, ultimately putting it in line with many other buildings assembled out of factory prefabricated standard elements. Many projects of this kind are found to this day, typically associated with low cost developments.

The disadvantage of such modular approach to reconfigurability is that along with increased ease of reconfiguration, the tightly integrated modules themselves constrain the freedom of adaptation on sub-module scale, thus hindering building's ability to adapt to circumstances that were not foreseen by designers of the building system and its modules.

A more generic approach facilitating reconfigurations within the building is the much earlier open plan concept originating from work of Louis Sullivan, Frank Lloyd Wright and Le Corbusier and since then commonly practiced throughout many modern buildings. Use of structural columns and allows to leave the building plan open. Although Corbusier's vision was for these spaces to remain undivided, an open plan approach allows easy insertion of spatial divisions and fast reconfigurations of spaces within the building, since such insertions don't have any structural properties. Contemporarily, many project developers provide dwellings to their clients without creating any internal partitions and allow the actual users to define how the flat, if at all, is to be subdivided, this signifies the shift from open plan to non-plan[1]. Some experimental projects like NEXT21's residential building in Osaka, Japan, allow not only for user-customized dwelling interiors, but also selection of customizable façade panels.

2.2. Adaptation by portability

As opposed to buildings permanently fixed to a site, there are multiple cases when human beings live their lives without being attached to one place. Historically, it applies to all cultures that have not progressed from hunter-gatherer to agrarian lifestyles, however also numerous contemporary groups and individuals choose for a way of life that is not bound to any permanent geographical location. In the twentieth century a new kind casual nomads have emerged; tourists. Camping has become as purely recreational activity.

Nomadic communities have always built shelters in ways that allow for their easy dismantling, transportation and deployment in a different location. Such portability of shelters can be achieved in many ways, however many similarities are shared between all of them.

In various cultures different types of portable shelters have emerged. Among them we can find all sorts of traditional structures such as chums, flys, loues, goahtis, lavvus, pandals, sibley tents, tarpaulins, tipis, wigwams, yurts and many others. Despite the diversity of their types, in all cases they contain a frame made of stronger material and sheets of fabric or

[1] Paul Barker, 'Non-plan Revisited: Or the Real Way Cities Grow' 12, no. 2, Journal of Design History (1999).

other material to provide cladding over that frame. Types of materials vary, depending on many factors such as availability of materials, environmental conditions and culture of their inhabitants. Structures of this kind can be easily assembled, dismounted and if necessary transported. Depending on the used technology, some or even all their parts can be left behind and reacquired in the new locations, while other parts can be reused multiple times, giving them thus often also certain capacity for re-configurability.

Contemporary tents use similar structural principles with modern materials and technologies. However, industrial production and standardization of their components, despite many unquestioned advantages such as lightness, resistance, capability to cover large spans and good insulation, has taken away their ability to be freely reconfigured. Spatial forms of a vast majority of tents cannot be easily changed or altered as it was the case with their more primitive predecessors.

Along with the progress of transportation technology, another approach towards creation of portable shelters has emerged, not requiring the disassembly and reassembly of building units for transportation. Caravans (travel trailers), campers (recreational vehicles) and prefabricated container buildings (mobile homes, static caravans, office containers, portable offices, mobile offices) allow for free relocation of intact entire structures, which are capable of either moving on their own, or are using other vehicles for towing or transporting them otherwise. Prefabricated container buildings often also allow for creating larger aggregated structures from many individual units.

Another group of portable shelters, which not only don't require disassembling for transportation, but are fully operational while on the move, are various sorts of covered boats. Their scale varies from small yachts to enormous cruise ships. In many ways small yachts can be compared to campers. Large ships, however, don't have any land-based counterparts, since water allows for easy relocation of much larger structures that can ever be moved on land.

Numerous examples are known of floating homes. They are in many ways similar to mobile homes, with the main difference being their placement on a floating base instead of firm attachment to solid ground. The concept of portability has always inspired experimental artists and architects. In 1960s Ron Herron came up with a concept of entire cities that move in his Walking Cities project. Over four decades later, a walking building was prototyped by a Danish group N55.

2.3. Adaptation by embedded flexibility

Examples presented so far require integral interference with built structure in order to achieve desired building adaptations, either by reconfiguring that structure or detaching it from its location. Such activities may happen periodically, however are not suitable when more frequent spatial adjustments are required.

Flexible building elements may be introduced when changes in spatial organization within a building happen regularly. The most common ones are doors of various types that allow for connecting and disconnecting separate rooms or other spaces when necessary. These kinds of flexible spatial divisions have been developed in most know building traditions, however often varied in technological solutions that were used. Most common are hinged doors of different scales that often replaced a simpler solution of a curtain dividing spaces. One of the most interesting techniques here is this found in traditional Japanese houses, where shoji are movable walls that slide sideways allowing for almost complete connection or separation between two parts of the house.

Contemporarily various sorts of doors, folding walls, curtains shades or shutters are commonly used in buildings to achieve quick transformation of certain spatial features or to adapt building spaces to changing conditions such as time of the day, weather or visibility to and from the outside. Usually those systems are applied on a small scale and have only two states; open and closed. Nevertheless there are numerous exceptions where buildings are equipped with much larger and more sophisticated flexible elements, such as embedded folding furniture, movable or foldable stairs, folding balconies and many other. In all cases it can be generalized that flexible building parts are introduced when there is need for frequent and repeatable reconfiguration between two or more states of a building space.

Simple flexible building elements such as doors, windows or shades are operated directly using the muscle power of building users. Technology allowed replacing human muscles with other kinds of energy. One sort that most pervasively found its use in building is electricity. Its first application was artificial lighting of buildings, which created one of the most revolutionary changes in humankind, allowing our living spaces almost full adaptability to external lighting conditions. Other applications followed, giving rise to the new field of building automation[1].

2.4. Adaptation by automation

Use of electricity in architecture led to emergence of the whole new kind of building adaptability: automation. Architectural automation by its definition requires implementation of a control system that intermediately steers flexible building features. Such control systems operate in connection to actuators and sensors. In architecture, actuators, sometimes also called effectors, are building components that perform actions that directly lead to certain spatial changes. Sensors, on the other hand, are elements that gather information from the environment and directly or indirectly control the actuators. In the simplest setup a sensor is directly connected to an actuator. Such is the case with a common light switch (sensor) that simply turns on or off the flow of electric current through the light bulb (actuator). However, in such case a control system doesn't exist, making it a flexible, but not automated system. Same is the case with automatic doors or electric blinds – if controlled with a simple switch sensor, they should be considered to be flexible, not automated building features. However, when we look at an elevator, it is already more obvious, that there exists a control mechanism that gathers input from all buttons, both inside the elevator and on all floors, as well as through other types of sensors is informed about current position of the elevator. Based on all this input the system determines when the elevator should go up and down, when it should stop and when to open or close its doors. Examples of this simple kind of automation are common on all scales. On Schowuburgplein in Rotterdam designed by West8 passers-by can switch positions of any of the four 15m high lighting elements using a simple console switchboard. The control system in this case takes care of extending and contracting selected actuators to reach configuration selected by the visitor. It also monitors the time passing between user inputs to prevent lighting elements from moving more often than once in every 30 minutes.

Even though electric devices of various sorts have dominated the field of building automation, other forms of sensing, control and actuation are also possible. Mechanical, pneumatic and hydraulic solutions are applied in many cases, often in combination with electric devices, but not necessarily.

An example of this may be one of the simplest and most common building automation appliances - a mechanical thermostat. There is always a set-point temperature that a thermostat is configured to have, which in fact is a threshold range between minimum

[1] Hermann Merz, Thomas Hansemann, and Christof Hübner, 'Introduction to Building Automation', in *Building Automation*, 2009, 1–25.

and maximum allowed temperature. When the surrounding temperature drops below the minimum value using a simple mechanical process, it opens up the valve leading in hot water or other heating matter to start the heating process. The valve closes again when the temperature sensed by device is higher than a maximum value in the allowed threshold. Such process is called a feedback loop, meaning that output of the system is constantly evaluated by the control system and if needed (counter) action is taken. As in the given example such feedback loops can operate using simple mechanical logics, however use of electricity, electronics and computer systems respectively increases amount of parameters that may be accounted for in such systems.

Modern home and building automation systems keep growing in complexity. Advancements in computer technology led to creation of powerful controllers capable of mapping a large variety of sensor input to diverse actuator output and types. Building control systems are usually centralized in residential houses and partly distributed in large buildings regardless of their function with prosepcts for further distributions being technologically investigated[1]. The controlled features commonly include: lighting, HVAC (heat, ventilation, air-conditioning), water system, security, surveillance, communication and entertainment. Actuators may include lights, display screens, alarms, vents, heaters, shading systems, signage, and many other specialist devices. Sensors can determine presence of people in specific areas of the building, smoke, temperature, humidity, amount of light and many other. Dedicated terminals or other interfaces may be used for manual input of other specific parameters if needed; data can also be acquired from building users using personal computers or mobile devices. Often these systems are function specific and not interconnected between each other; however current trend is for increased integration.

Looking from the large perspective of building adaptability, two main points of concern may be raised in respect to building automation. The first is that building automation systems are very rarely integrated with building architecture. Normally they are seen as add-on installations to traditionally designed building form. This can be most clearly seen in the house of Bill Gates, which is often quoted as the most advanced example of domotics (home automation). It is by design a large, yet traditional Pacific Lodge villa, which has been equipped with an overwhelming amount of technological features. A counter example is this of Werner Sobek's house R128, which due to applied automation technology is a very eco-friendly building and where building installations have been (yet arguably) integrated with building's design.

The second, much more profound problem with building automation is the linearity of embedded system logics. Regardless of control systems' complexity, their logics are normally constructed in a tree-like manner, using chains of conditional statements (if X then Y else Z). Although building automation is inherently flexible and responds to a very wide range of dynamic factors, it is only the procedures that have been predicted by system designers that the system may perform. Building automation works as long as users use the building exactly as it was designed to be used and if external environment behaves in the originally predicted way. Reprogramming the way in which building system operates, if at all possible, usually has to be performed by highly specialized personnel. Therefore overall adaptability of automated buildings, if defined as ability to adapt to unforeseen conditions, is usually very low. On the other hand, the frequency of local adaptations to specific predictable factors can be very high.

[1] Andrzej Jablonski, Ryszard Klempous, and Benedykt Licznerski, 'Diversified Approach to Methodology and Technology in Distributed Intelligent Building Systems', in *Computer Aided Systems Theory - EUROCAST 2003*, 2003, 174–184.

2.5. Spatial adaptation throughout scales

The four described types of spatial adaptation are not mutually exclusive. Different aspects of one building system may be adapted using different strategies. Usually any system can be in some way reconfigurable. Implementation of features allowing for portability, flexibility or automation occurs when there is high probability of certain reconfigurations to be required in a recurrent manner.

Clearly, spatial adaptations require least resources (time, energy, material) when performed on small scales. Large scale adaptations can be achieved radically, by transforming the entire system, or through a high number of small scale, local adaptations. As large scale whole-system adaptation are slow and resource demanding, achieving large scale adaptations through accumulation of small scale adaptations offers a more efficient alternative, although radical whole-system transformations may be more difficult to perform in this manner.

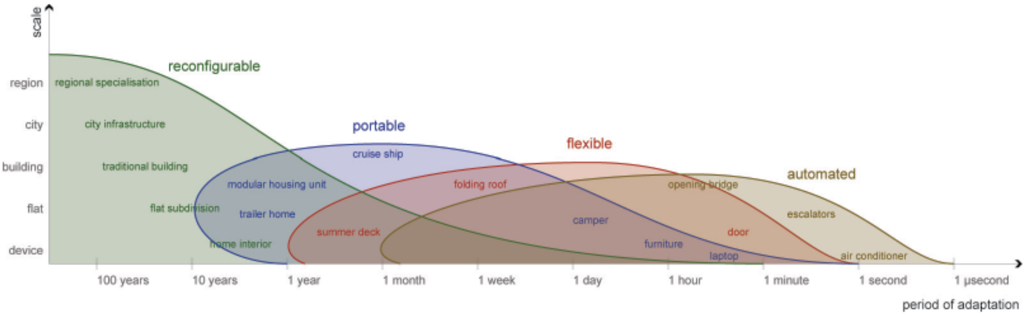


Fig.8. Generally observed relation between specified types of architectural adaptation and affected architectural scale.

3. Envisioning dynamic architecture

Although architecture in its essence has always been a dynamic process, understanding and creating buildings as dynamic systems requires a cultural and technological shift. First ideas leading to this way of understanding buildings can be traced back to the industrial revolution and following it "machine age".

3.1. Buildings are machines

In the end of the 19th century the western world has been deeply transformed by the effects of the industrial revolution. Prefabrication and mass production allowed for rapid technological development, at the same time disrupting the social structure and culture, not necessarily in a desired way, as it created inhumane working conditions for factory labourers and unprecedented pollution. However, it has also triggered rapid appearance of new, revolutionising inventions such as light bulbs, telephones, automobiles, airships, or elevators, many of which had an enormous impact on architecture. It became clear that architects needed to develop a new approach for creation of living and working spaces that would take advantage of these new technologies and at the same time accommodate the new culture of the machine age with all its opportunities and threats.

Although the statement that "houses are machines for living in" (1924) belongs to Le Corbusier, he has only popularised the idea that has already been addressed by other, preceding him, avant-garde architects of the time. One of them was as an Italian futurist Antonio Sant'Elia who wrote of buildings that would be "similar to a gigantic machine", leading to "architecture as new as our frame of mind is new", that should be achieved "with the aid of every scientific and technical resource"[1] (1914). German Bauhaus members or Russian Constructivists had been pursuing similar visions, regardless of differences in routes taken to achieve them.

It is also Le Corbusier who in his manifesto[2] (1924) propagated the concept of the open plan - a floor plan where columns are introduced to entirely replace obstructing structural walls allowing flexible subdivision of building space and ability to adapt it to a wide range of functions. This idea had originally been formulated in the process of creation of the first skyscrapers by Louis Sullivan (1890). Newly invented elevators allowed stacking multiple floors above each other and in result of this, buildings occupying a small footprint and sharing one entrance could have had an exponentially larger floorspace and thus host multiple, frequently changing tenants with changing layout requirements. For this, an open plan was an ideal solution. This idea has been further explored and popularised by Sullivan's apprentice Frank Lloyd Wright, among others in his residential projects of "Prairie Houses" (1900-1917) that later inspired le Corbusier. The concept of the open plan went beyond the idea of preset efficiency in building configurations; it demonstrated that in order to be efficient, way in which buildings function needs to change over time. An illustrative and visionary example of an open plan skyscraper of early modern times is the far-sighted Glass Scyscraper project (1921-22) of Mies van der Rohe, who ultimately realised an altered version of his glass skyscraper dream (which has lost almost entirely its visionary appeal in the process) in 1958 together with Philip Johnson in the Seagram Building in New York, which became an archetype of a modern high-rise.

Despite the open plan idea allowing flexible reconfigurations of internal building spaces, most modernist visions have been attempting to define fixed functional plan arrangements that would create well engineered, yet static configurations of spaces, ergonomically designed for typical activities of their users and efficiently fabricated in terms of cost, time and material. Alongside other iconic projects of that era an illustration of this trend may be the early work of Buckminster Fuller, who ahead of his time had been trying to create buildings such as the Dymaxion House (1927), that was an assembly of cheaply prefabricated, yet at that time state-of-the-art engineered components including entire kitchen and bathroom units.

3.2. Adaptive cities

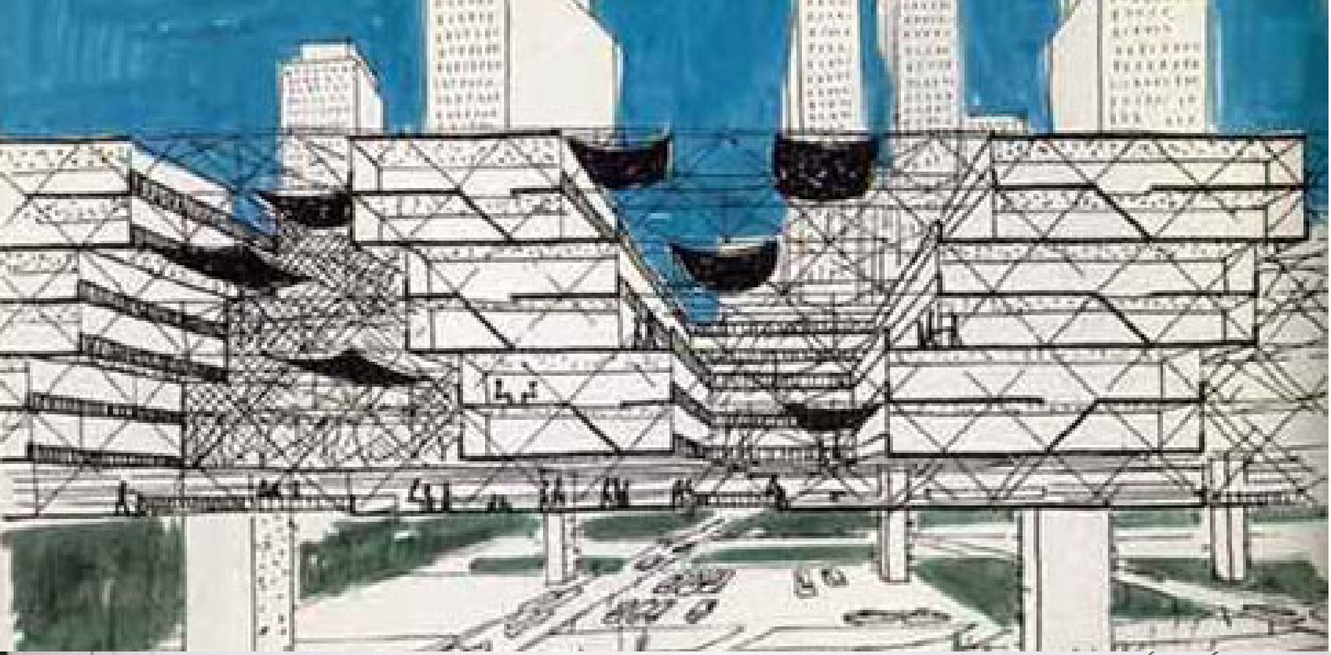
The prefabrication of building components had a big impact on post-war architecture, allowing for rapid rebuilding of housing after the immense destructions of the Second World War. Whether in case of prefabricated detached houses of American suburbia, west European modernism following examples set by Corbusier's Unité d'Habitation or prefabricated housing units popular in the communist eastern block countries, same problems prevailed. Created housing has become highly repetitive and monotonous, offering dull living environments that inhabitants felt disconnected from, often leading to generation of numerous social pathologies.

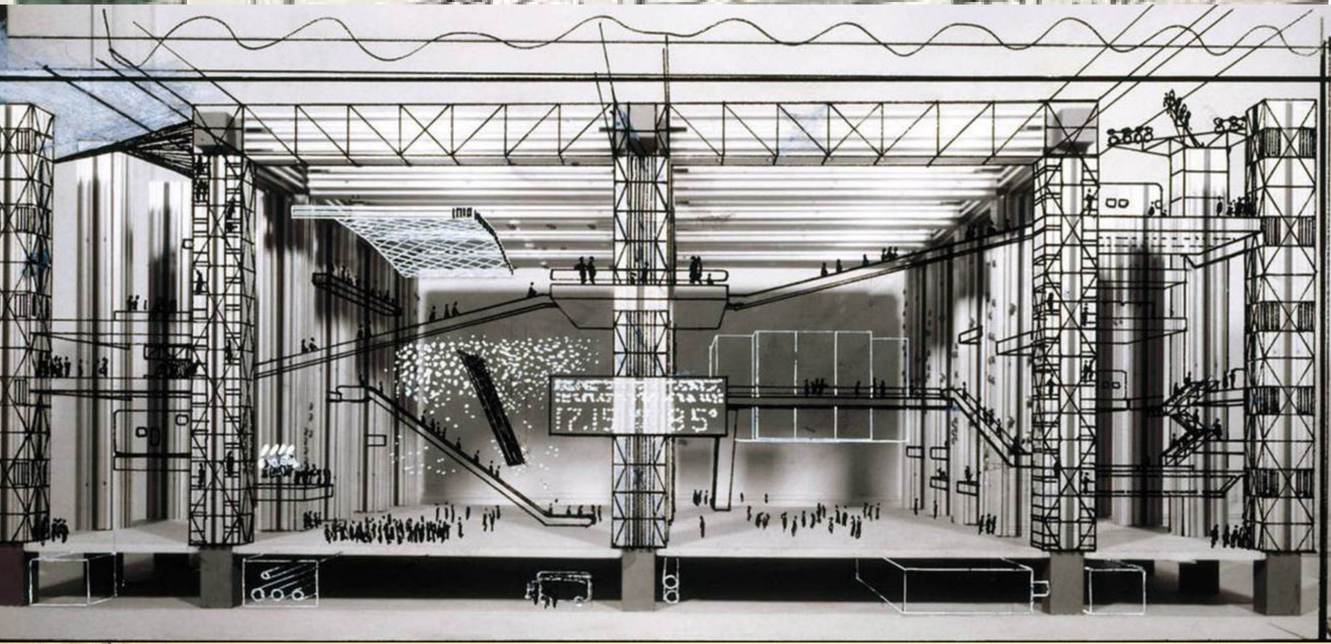
A public reaction to this "prefabricated dullness" was either a regressive resurrection of traditional architectural solutions, especially fuelled by the post-modern period in architecture, or attempts in finding solutions that would introduce flexibility and diversity to architecture without giving up the technological advancement of modern architecture.

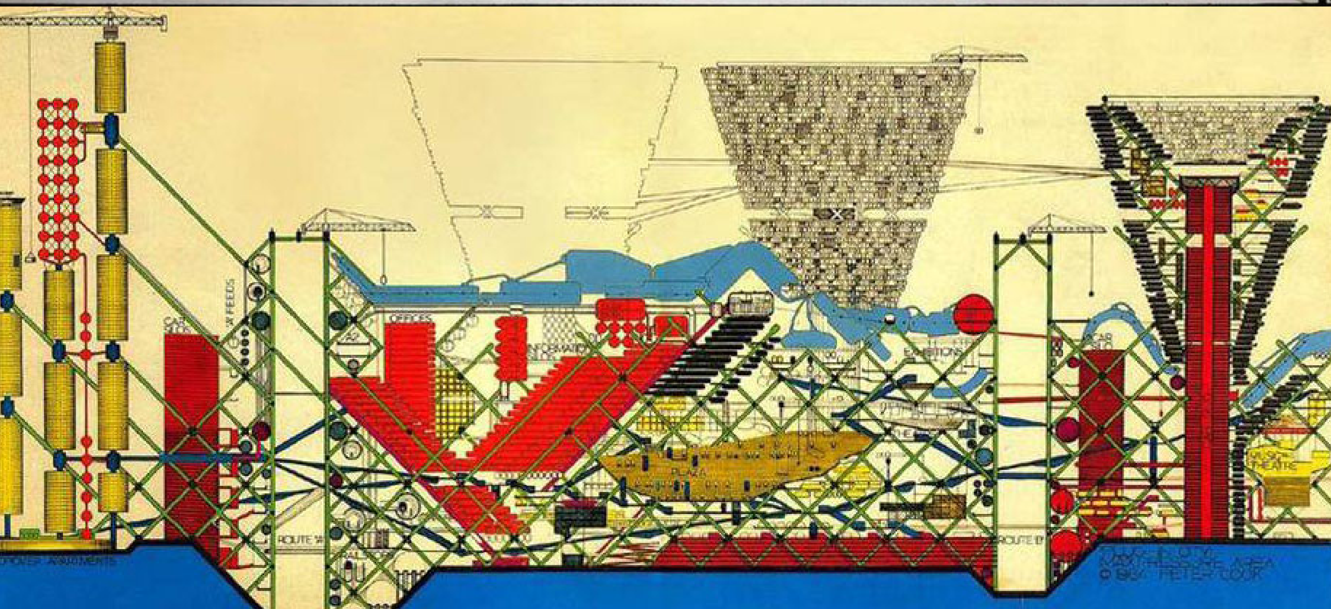
Premises of the concept of an open plan have been extended to the urban scale in the vision of Yona Friedman. In his manifesto from 1958 entitled "mobile architecture" he proposes "dwelling decided on by occupant" and "infrastructures that are neither determined nor determining".

[1] Ulrich Conrads, *Programs and Manifestoes of 20th-Century Architecture* (The MIT Press, 1975).

[2] Ibid.




17.15 85°


OFFICES
INFORMATION
EXHIBITIONS
PLAZA
MUSIC THEATRE
FEEDS
ROUTE
© 1964 PETER COOK

His “Spatial City” project illustrates this idea. A whole city is created on a framework of a modular grid truss. This structure provides structural support, power, water and a sewer system. The structure can be modified and expanded over time, various building and infrastructure modules can be inserted into it. The concept of Friedman builds upon many Modernist ideas, such as detachment of buildings from the ground, minimalist formal appeal or prevalence of grid organisation, while it also questions the lack of consideration for individuality of the inhabitants of architecture. In response Friedman proposes an environment where the role of an architect is reduced to being a facilitator, creating an open system permitting free reconfiguration expressing changing spatial needs of individual users of proposed urban spaces.

The work of the Metabolists includes many similar ideas, developed in post-war Japan, troubled with severe housing shortage at that time. In their vision, a city resembles a living organism that should be allowed to expand in a way resembling organic growth. Projects like Kenzo Tange’s planning scheme for Tokyo Bay or Kiyonoiri Kikutake’s Marine City (1958-63) provided lucid illustrations of their ideas. The Nagakin Capsule tower (1972) has become the only built project of the Metabolist movement. It consists of 140 prefabricated, self-contained capsules attached to a central core. Despite a theoretical possibility, the initial arrangement has never been altered, although since the building has fallen into disrepair, replacement of capsule was being considered in 2007, however not executed to date.

On the American front, similar concepts have been realised by Moshe Safdie in the Habitat 67 project built for the Expo in 1967. However, in this project, rather than prioritising the ability to grow of the built environment over time, underlined other qualities of proposed system, which allowed a high diversity and quality of living spaces (each housing unit had a garden) at an affordable price, while maintaining high density of the development.

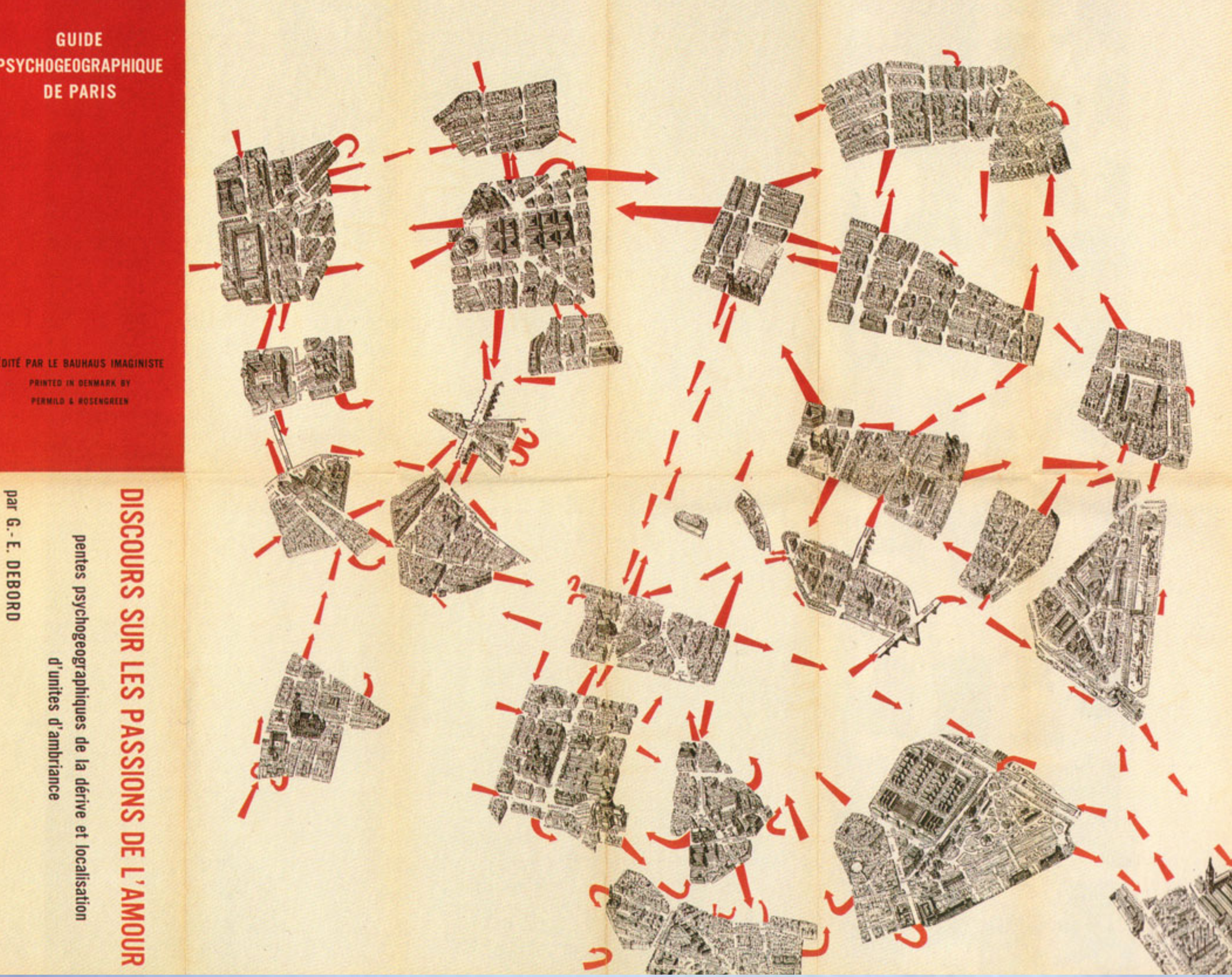
The projects of Archigram group took the architecture - machine analogy to a further level. Their conceptual projects have been published in a series of pampflets also entitled “Archigram”. In their view Modern architecture has lost its “guts” and didn’t hold true anymore to its original vision. Proposed projects were of different scale, from small living capsules to entire cities. In all cases latest technology of the time, including space-age devices were used to create living environments allowing high mobility and flexibility. Diverse kinds of transportable and deployable capsules were introduced, such as the “Living Pod”. In city-scale projects such as “Plug-in City” or “Seaside Bubbles”, infrastructure has been provided to facilitate flexible assembly of diverse units into entire urban systems. In “Instant City” large spatial environments would move lifted by airships and travel between locations, producing new events and situations. In David Greene’s words, projects of Archigram “(...) nomadism is the dominant social force; where time, exchange and metamorphosis replace stasis; where consumption, lifestyle and transience become the programme; and where the public realm is an electronic surface enclosing the globe.”[1] Projects of Archigram took technology to its limits, in a playful way showing its potential to create dynamic habitats for people.

A distinct line of concepts has emerged from the founded in 1957 counter-capitalist Situationists International movement[2]. Situationists experimented with “the construction of situations” allowing individuals to pursue their own, primitive desires. Their concept of *unitary urbanism* postulated rejection of Euclidean and functional approach to architectural and urban design and integration of art in daily life. The maps of Guy Lebord illustrate those concepts, emphasizing subjectivity of spatial urban experience. In partial opposition to Lebord, whose focus was oriented towards the “content” of spatial experience, Constant Nieuwenhuys concentrated in his ideas on the spatial infrastructure that would be suitable

[1] Dennis Crompton, *Concerning Archigram*, New edition (Archigram Archives, 1999).

[2] Ken Knabb, *Situationist International Anthology*, Revised & Expanded (Bureau Of Public Secrets, 2007).

Img. 2. Left, from top: Yona Friedman “Spatial City”, Cedric Price “Generator”, Peter Cook “Plug-in City”


GUIDE
PSYCHOGEOGRAPHIQUE
DE PARIS
ÉDITÉ PAR LE BAUHAUS IMAGINISTE
PRINTED IN DENMARK BY
PERMILD & ROSENGREEN
DISCOURS SUR LES PASSIONS DE L'AMOUR
pentes psychogeographiques de la dérive et localisation
d'unites d'ambriance
par G.-E. DEBORD

as a container for the Situationist city. His project New Babylon, on which he worked on from mid-1950s until late 1960s, is the manifestation of these endeavours. Constant wrote in 1966; “What is New Babylon actually? Is it a social utopia? An urban architectural design? An artistic vision? A cultural revolution? A technical conquest? A solution of the practical problems of the industrial age? (...) Each of these questions touches an aspect of New Babylon”[1]. In New Babylon the user of Space is Johan Huizinga’s “Homo Ludens”: Man the Player, he or she is the free and creative nomad who takes space into his own hands. Around this idea a structure of the city is formed, being the network space built up of sectors, which “grow” around social networks of individuals. As Constant writes: “the ludic life of the inhabitants of New Babylon presupposes frequent transformation of the interior of the sectors. For this to take place without problems, the containing structure would have to be as neutral as possible, and, from the construction point of view, the variable contained structure [would have to be] completely independent of the former”[2]. The new Babylon has been an entirely theoretical project, not intended to ever be built, especially by Constant himself who spent later years of his life criticising the project and pointing out the dangers of a ludic society. However, its ideas have remained highly influential for architectural discourse and became rediscovered for contemporary architectural discourse by Marc Wigley’s in “Constant’s New Babylon, the Hyper-Architecture of Desire” published in 1999[3] and “The Activist Drawing: Retracing Situationist Architectures from Constant’s New Babylon to Beyond”, edited with Catherine de Zegher in 2001[4].

Constant’s vision has illustrated the interplay between a utopian, ludic society and the hypothetical space evolving around it. It did not put much attention into any actual technology that would be needed to create such space capable of continuous adaptation, assuming that its users would gradually transform the space they inhabit. Such acts of transformation would not be seen as work, but more a game being played by New Babylon inhabitants.

On the other hand, ideas of Archigram provided more technology-inclined illustrations. Structural systems, cranes and other space-age devices are employed to construct and re-construct the plug-in city and all other projects. Nevertheless, in both cases, architecture is directly controlled by its users. It does not have any autonomy.

The project which stands out in respect to its introduction of architectural agency is Cedric Price’s Fun Palace, which became an icon of Interactive Architecture[5]. In 1964 Price proposed a building whose “form and structure, resembling a large shipyard in which enclosures such as theatres, cinemas, restaurants, workshops, rally areas, can be assembled, moved, re-arranged and scrapped continuously”, in which you can “choose what you want to do – or watch someone else doing it. Learn how to handle tools, paint, babies, machinery, or just listen to your favourite tune. Dance, talk or be lifted up to where you can see how other people make things work. Sit out over space with a drink and tune in to what’s happening elsewhere in the city. Try starting a riot or beginning a painting – or just lie back and stare at the sky”. The concepts proposed in the Fun Palace were truly revolutionary for their time. The project proposed to allow its users to freely modify the organisation of its space. At the same time, user activities would be monitored and future behaviour and organisation of space would autonomously try to adjust to previously acquired knowledge about user preferences in connection to specific situations, ultimately leading to creation of space interacting with rather than being controlled by its users. Even though not realised, Price’s vision crystallised

[1] Mark Wigley, *Constant’s New Babylon: The Hyper-architecture of Desire* (010 Publishers, 1998).

[2] Constant Nieuwenhuys, ‘New Babylon’ (Haags Gemeetenmuseum, 1974).

[3] Wigley, *Constant’s New Babylon: The Hyper-architecture of Desire*.

[4] Catherine de Zegher and Mark Wigley, *The Activist Drawing: Retracing Situationist Architectures from Constant’s New Babylon to Beyond*, illustrated edition (The MIT Press, 2001).

[5] Stanley Mathews, *From Agit-prop to Free Space: The Architecture of Cedric Price*, 1st ed. (Black Dog Publishing, 2007).

Img. 3. Left, from top: Guy Debord “Psychogeographic guide to Paris” cover, Constant Nieuwenhuijs: “New Babylon” a new notion in architectural discourse. The idea of “non-plan” became a more ambitious successor of the “open plan”. It proposed the users and their bottom-up activities as a replacement for the authoritarian top-down planning defining every aspect of space on all scales. The role of “fun palace” is also significant, as it is a seminal example of architectural application of cybernetic theories.

3.3. Architectural cybernetics

The mid-century explosion of projects involving a more progressive view on architectural adaptation have a strong connection to the science of cybernetics, which has actively developed around the same time. The foundations of cybernetics have been laid by Rosenblueth, Wiener and Bigelow In their historic paper, “Behavior, Purpose and Teleology”[1]. Ideas presented there stem from considering an object and its environment through their mutual relations and the term “feedback” becomes for the first time defined, along with the concepts of “non-purposeful” and “purposeful” behaviour, subdivided further into “predictive” and “non-predictive”. In following works of Wiener, especially in “Cybernetics”[2], which defined the name for the entire domain, these ideas are elaborated further and concepts of “complexity” and “self-organisation” of systems are presented. Warren Brodey discussed application of cybernetic concepts to architecture in his essay “Soft Architecture – The Design of Intelligent Environments”[3], most noteworthy the idea of adaptability[4] of architecture through its co-evolution with human behaviour, with man and its surrounding being “both object and environment to each other” along with many explicit examples. Similarly, Andrew Rabneck proposed the application of cybernetic technology to achieve adaptation in architecture[5] .

An important role in formulating the technical side of the emerging vision of architecture capable of autonomous interaction with its users belongs to the cybernetist Gordon Pask. John Frazer writes; “Architectural thinking in the sixties was preoccupied with issues of flexibility, impermanence, prefabrication, computers, robotics, and a global approach to energy, resources and culture. The implied systems thinking in architecture inevitably came to embrace cybernetics and cybernetics in architecture inevitably came to embrace Gordon Pask“[6]. Gordon Pask became one of the most influential figures in the discourse about creation of interactive systems and interactive architecture as a specific kind of those. He became mostly known for his “conversation theory”[7] in which he describes how interactions lead to creation of knowledge. His other theory on “interactions of actors”, is focused at a broader perspective, it presents a worldview where all “processes produce products and all products (finite, bounded, coherent objects) are produced by processes”. His work suggests that our entire universe may be seen as a highly complex system of interacting agents, where everything that’s measurable is a product (and cause) of those interactions. Pask’s view on architecture is concisely described in his article “Architectural relevance of cybernetics”[8] where he not only discusses the evolution of architecture-inhabitant systems,

[1] Arturo Rosenblueth, Norbert Wiener, and Julian Bigelow, ‘Behavior, Purpose and Teleology’, *Philosophy of Science* 10, no. 1 (1943): 18–24.

[2] Norbert Wiener, *Cybernetics, Second Edition: Or the Control and Communication in the Animal and the Machine*, 2nd ed. (The MIT Press, 1965).

[3] Warren Brodey, ‘The Design of Intelligent Environments: Soft Architecture’, *Landscape* 17, no. 1 (1967).

[4] Francis Heylighen, ‘The Science Of Self-Organization And Adaptivity’, *Organizational Intelligence and Learning, and Complexity, in: The Encyclopedia of Life Support Systems (EOLSS)* (1999): 253–280.

[5] Andrew Rabeneck, ‘Cybermation: A Useful Dream’, *Architectural Design* (1969).

[6] John Frazer, ‘The Architectural Relevance of Cybernetics’, *Systems Research* 10, no. 3 (1993): 43–48.

[7] Gordon Pask, *Conversation Theory, Applications in Education and Epistemology*, 1976.

[8] Gordon Pask, ‘The Architectural Relevance of Cybernetics’, *Architectural Design* (1969).

but also architect's role in it. Several models were constructed to conceptualise operation of architectural adaptation driven by cybernetic principles. The most renowned belong to Yona Friedman[1] and Charles Eastman[2], the first one proposing a centralised feedback loop model and the second a fully distributed system. In the same time Goldman proposed a more elaborate model directly inspired by complex biological processes. He argues that from the cybernetic standpoint "buildings are homeostatic systems, which, due to their functional behavior, may be classified as living systems"[3].

Next to theoretical models late 1960s also became the time of many experiments, where said systems were attempted to be built. Nicolas Schöffer became was the precursor of cybernetic art, with his first installation built in 1956, being based on Wiener's idea of a feedback loop, utilising different media including light and sound positioned in space and actively involving spectators. Many other artworks followed. Pask's theories may be well illustrated by his own early installation "The Colloquy of Mobiles" made for the "Cybernetic Serendipity" exhibition held at the ICA in London, in 1968. In this installation five objects of two kinds were suspended from the ceiling. Two "males" were equipped with light sources, three "females" they all communicated with each other using light and sound. After a pause of inactivity, females' bodies would begin to glow, attracting attention of males. Males would then shine a beam of light towards females and females would try to deflect this beam towards the light sensors of males. If achieved, this would give them a mutual moment of "satisfaction". Using feedback loops and learning from past experience, agents of the installation would learn over time how to optimise their behaviour in order to shorten the time and amount of energy needed to achieve that state.

Cybernetics and related complex system theories remained very influential for many later architectural experiments. Nicholas Negroponte in his two books "The Architecture Machine: Toward a More Human Environment"[4] (1973) and its follow up "Soft Architecture Machines"[5] (1975) explores application of emerging at that time computer technologies to design and creation of architecture. In the introduction to "Soft Architecture Machine", Gordon Pask summarizes the cybernetic underpinnings of ideas further presented by Negroponte. Negroponte argues for development of artificial intelligence to be embedded in buildings, he also explores possibilities of new computer aided design tools, introducing the notion of "computer-aided participatory design" (guest-introduced by Yona Friedman). One of Negroponte's experiments called SEEK features living gerbils placed in an environment made up of 500 two-inch metal plated cubes. Whenever gerbils would slightly dislocate any of the cubes, it would then be moved by a robotic arm to a new location and realigned to a grid. In this way a feedback loop was established between curious gerbils and a machine system building up the space those gerbils occupied.

The systemic aspect of a self-aware architectural space has further been advanced by John Frazer, published in "An Evolutionary Architecture"[6]. Gordon Pask who has also provided an introduction to this work wrote about the new role of an architect, which role "is not so much to design a building or city as to catalyse them; to act that they may evolve". One of presented Frazer's projects, "The Universal Constructor" (1990) is a model similar to Negroponte's SEEK,

[1] Yona Friedman, 'Information Processes for Participatory Design', in *Design Participation. Proceedings of the Design Research Society's Conference* (Manchester: Academy Editions, 1972).

[2] Charles Eastman, 'Adaptive-conditional Architecture', in *Design Participation, Proceedings of the Design Research Society's Conference* (Manchester: Academy Editions, 1972), 51–57.

[3] John A. Goldman, 'Towards a Systems Biology of the Built Environment', *Energy and Buildings* 2, no. 4 (December 1979): 309–321.

[4] Nicholas Negroponte, *The Architecture Machine: Toward a More Human Environment* (The MIT Press, 1973).

[5] Nicholas Negroponte, *Soft Architecture Machines* (The MIT Press, 1976).

[6] John Frazer, *An Evolutionary Architecture* (Architectural Association Publications, 1995).

however the difference lies in radically different approach to the automation of model reconfigurations. Each block contains a microcontroller and can communicate to other blocks that are stacked above and below. It can also send signals to humans interacting with the model "using lights the model (…) indicates its proposed response by asking the interactor's assistance in adding or removing units". In the second part of his book, Frazer looks into computation strategies such as neural networks and genetic algorithms that would allow improving the embedded intelligence of such self-reconfiguring models.

Contemporarily cybernetics has dissolved as a distinct field, spreading into numerous application-specific branches, with the newly emerged systems science attempting to reconsolidate the more theoretical endeavours. From late 1970s until 1990s, despite its great promises, the interest in architectural cybernetics has declined, giving place to the anti-technological early postmodern architecture. Nevertheless, the profound influence of cybernetics on the way of thinking about architectural adaptation remains undisputed.

3.4. Ubiquitous computing

Many of the ideas theorised by cyberneticists found their continuation in the domain of ubiquitous computing. As Mark Weiser defined it in 1991, "Ubiquitous computing names the third wave in computing, just now beginning. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop. Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives."[1], forming "a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network"[2]. Often interchangeably with "ubiquitous computing", terms "pervasive computing" or "ambient intelligence" have been used[3], despite subtle differences in meaning. The term "internet of things" denotes a more recent and broader concept defined, among others, in the official EU strategy as "a network of interconnected objects, from books to cars, from electrical appliances to food, and thus (…) an 'Internet of things' (IoT). These objects will sometimes have their own Internet Protocol addresses, be embedded in complex systems and use sensors to obtain information from their environment (e.g. food products that record the temperature along the supply chain) and/or use actuators to interact with it (e.g. air conditioning valves that react to the presence of people). The scope of IoT applications is expected to greatly contribute to addressing today's societal challenges: health monitoring systems will help meet the challenges of an ageing society; connected trees will help fight deforestation; connected cars will help reduce traffic congestion and improve their recyclability, thus reducing their carbon footprint. This interconnection of physical objects is expected to amplify the profound effects that large-scale networked communications are having on our society, gradually resulting in a genuine paradigm shift."[4]

[1] John Krumm, *Ubiquitous Computing Fundamentals* (Chapman and Hall/CRC, 2009).

[2] Mark Weiser, R. Gold, and J.S. Brown, 'The Origins of Ubiquitous Computing Research at PARC in the Late 1980s', *IBM Systems Journal* 38, no. 4 (1999).

[3] Daniel Ronzani, 'The Battle of Concepts: Ubiquitous Computing, Pervasive Computing and Ambient Intelligence in Mass Media', *Ubiquitous Computing and Communication Journal* 4, no. 3 (2009).

[4] 'Internet of Things - An Action Plan for Europe, Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions' (Commission of the European Communities, 2009), http://ec.europa.eu/information_society/policy/rfid/documents/commiot2009.pdf.

Numerous experiments of network of autonomous devices operating in human living environment have been performed under umbrella terms referred to to as “smart environments” or “intelligent environments”. As an example, the “Adaptive Home” project’s premise was to create a living environment, which ”programs itself by observing the lifestyle and desires of the inhabitants and learning to anticipate their needs”[1], with features such as invisible interfaces and pro-active behaviour of the system developed in the course of the experiment[2]. Similar functionality has been developed within MIT’s “adaptive room” allowing its inhabitants “to interact with computational systems the way they would with other people: via gesture, voice, movement, and context” [3]. Numerous other experimental systems[4] have been developed within public and commercial research initiatives and many commercially available systems exist allowing extensible building automation[5]. However, in all cases these systems are introduced as additions to otherwise static buildings and have not been integrated with creation of spatial conditions. Their functionality is mostly limited to communication, control of entertainment systems, lighting, heat, air-conditioning and ventilation, security, and other aspects of the environment, yet not the spatial organisation of the environment in which they operate.

3.5. Architecture as medium

Despite the rapid development of information technologies in the second half of the 20th century, the mainstream architecture has not become strongly affected by the cybernetic concepts or following them developments of ubiquitous computing. Nicholas Negroponte ends Soft Architecture Machine with a warning that application new computer technology calls for a thorough revision of architectural methodologies, as it completely changes the kinds of problems that architects and users of architecture face in the digital age. This warning seemed unheard by the mainstream architectural practice. It seems that to this date mainstream architecture and digital technology have not converged to provide new qualities.

In 1972 Robert Venturi has published “Learning from Las Vegas”, which became one of the seminal texts for the postmodernism in architecture. His text may be accounted for re-discovery of architecture as a medium. In this way of looking at the communication between users and architecture, architecture is seen as a medium used to convey messages prescribed by its owners or designers. It stands in opposition to what Price, Friedman[6], Negroponte were suggesting in their projects, where architecture clearly became a communicating subject in itself.

Venturi has distinguished to types of architectural symbolism found along Las Vegas Strip, which he named “decorated shed” and “duck”, implying the distinction between on one hand signage attached to an otherwise meaningless “shed” and on the other hand; a structure that in carries a symbolic meaning as whole, akin to a poultry restaurant in Vegas shaped as a giant duck. The “decorated shed” approach seemed to have dominated throughout the postmodernism, whether the symbolic layer were the props referencing to classical architectural details or advertising neon lights and billboards. This happened to the extent

[1] M. C. Mozer, ‘An Intelligent Environment Must Be Adaptive’, *IEEE Intelligent Systems and Their Applications* 14, no. 2 (1999).

[2] M. C. Mozer, ‘Lessons from an Adaptive House’, in *Smart Environments: Technologies, Protocols, and Applications*, ed. D. Cook and Das (Hoboken, NJ: J. Wiley & Sons, 2005).

[3] Michael Coen, ‘Design Principles for Intelligent Environments’, in *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (presented at the AAAI’98, Madison, WI, 1998), 547–554.

[4] Diane Cook and Sajal Das, *Smart Environments: Technology, Protocols and Applications* (Wiley-Interscience, 2004).

[5] Frances Aldrich, ‘Smart Homes: Past, Present and Future’, in *Inside the Smart Home*, 2003, 17–39.

[6] Yona Friedman, *Yona Friedman: Pro Domo* (Actar, 2006).

that the 1981 revitalisation of Times Square in New York introduced a planning regulation obliging major buildings around the square to cover their facades in electronic signage. Similar phenomena took place in London's Piccadily Circus and Tokyo's Shibuya. Whether the effect achieved from the perspective of rejuvenation of the urban space may be disputed upon, it is clear that such approach creates an utter divergence between what architecture contains and what it communicates to its users.

The trend of media facades integrates attached signage into building facades, reducing this dichotomy. One of the interesting examples may be the Potsdamer Plats facade in Berlin by realities:united, where individually controlled TL lights were inserted in a partly irregular pattern, or the KPN building by Renzo Piano in Rotterdam where an array of large green on-off pixels is integrated into a flat facade. Recent projects introduce facade elements that can not only change colour, but can also deform kinetically(e.g. Flare), as well as projections on buildings that use optical illusions in relation to building details they are projected upon to create stunning visual effects (e.g Urbanscreen's 555Kubik). Nevertheless, the principles by which such facades culturally (and technologically) operate is no different than this of large screens attached onto buildings. Displayed typically visuals are prepared in advance and played in a loop on those facades without any feedback from spectators. Facades like this of Dexxia Tower in Brussels, Belgium are still rare and shy examples of interaction with passers-by. Control panel placed in the public space in front of the tower, designed by the group Lab[au], allows anyone to change the displayed animation by dragging new shapes onto the screen. Yet, even in this example, it is a control system made open to the public, rather than an introduction of any sort of interactivity between the building and its users.

Perhaps a better example may be the work of Graffiti Research Lab, an activist group, which among other things sets up large projections onto existing buildings. In their philosophy anyone should be allowed to do so, as public space is shared property of the society. Their activities include guerrilla movie projections or projections combined with tracking of a laser pointer, where anyone can "draw" anything directly on the building. Yet also in this case, building facade is reduced to a medium. Numerous art projects, such as "Sniff" by Karolina Sobecka or "Hand from Above" by Chris o'Shea introduce much more involving and entertaining relations between passers-by and urban screens. Many other art installations more openly address the role of new media in relation to architectural spaces. Clearly, a new branch of the language of new media1 is being distinctly developed in architectural domain.

3.6. Virtual architecture

In his 1984 novel "Neuromancer", William Gibson has popularised the term "cyberspace" – an abstract space of data communication. In the novel, a computer hacker character named Case uses a device called "deck" that directly connects to his brain and in this way provides him access to the virtual representation of the cyberspace matrix. Although cyberspace means a complex network of communication and data exchange, in Neuromancer it has a visual, three dimensional spatial manifestation that makes it appear as a kind of autonomous universe that is parallel to the one which is physically inhabited by Case. It is in cyberspace that Case ultimately encounters Wintermute, a superior artificial intelligence being that inhabits the network analogous to today's internet.

It had not been a long wait since then for the developments in computer technology to gradually allow generation of three dimensional virtual spaces that although can't yet be directly projected into our brains, may provide experiences comparable to the one of Gibson's cyberspace. Technologies originally developed for depiction of engineering models,

[1] Lev Manovich, *The Language of New Media* (The MIT Press, 2002).

ultimately found wide and rapidly growing application to computer entertainment and games. First three dimensional games were one-person experiences, but the convergence of 3d worlds with first local area networks and later the internet has spun off new genres of multi-player games that allow several players to compete over the network. Multi-player games have consequently evolved further into massive multi-player online games (MMOG) that can accommodate immense numbers of users. The score of the highest number of simultaneous players participating simultaneously in one 3d world keeps increasing, with the record as of December 2009 belonging to EVE online with over 54 000 simultaneous players reported and World of Warcraft having over 12 million users subscribed on a monthly base. Other platforms such as Second Life provide similar experience with the sole purpose of facilitating social interactions rather than gaming. In Second Life there is no narrative, competitions or goals such as in MMO games, players connect to Second Life e to meet each other, talk and explore together the 3d world that they can co-create themselves. Both MMO game worlds and Second Life have developed their own economies. Users can buy plots of virtual land as well as exchange or trade self-created virtual objects including virtual architectural elements.

In the 1990s architectural discourse has become influenced by the concept of cyberspace. Marcos Novak can be seen as the initiator of that discourse in his seminal text "Liquid architectures in cyberspace"[1]. Much of the popular debate became additionally fuelled by the 1999 release of Wachowski brothers' film "The Matrix" that presented a more contemporary vision of cyberspace to the popular audience and created an increase of the public awareness of the cyberspace concept. In that time innovative architects such as Gregg Lynn have been discovering the application of 3d design and animation software to architectural design and new possibilities that such tools create for architecture[2]. Asymptote of Hani Rashid and Lise Anne Couture is known for their early project of creating a virtual 3d world for the New York Stock Exchange trading area and the Virtual Guggenheim museum space. Creation of such virtual worlds allowed architects to work in an environment free of physical constrains. Many others followed, including numerous architects doing research into possibilities of creation of interactive architectural spaces into environments such as Second Life launched in 2003 and its predecessor Active Worlds which was operational already in 1995.

The superficially simple concept of the "virtual", however, demands further investigation to provide foundations for its implications in the next chapters. Following Marcos Novak's argument; "we, the finite, can never conduct a full critique of the infinite. Hence, by a curious reversal of terms, it is reality itself that is the most virtual for us, in the sense of being an asympotic potentiality that we can never fully know or exhaust. The 'real' is already virtual. As with the quantum universe, the difference between real and virtual is stochastic: a matter of probabilities. In a conventional sense, the real is that which is most likely. Technology is allowing us to change the common structure of probabilities and to stabilize formations that were previously so unlikely as to be delegated to the realm of dreams and miracles. Pure virtual space is the liquid structure of potentialities of all possible worlds, the quantum world, or the world that Italo Calvino's amazing character QwfwQ exists in, the rhizomatic world-before-categories where everything relates to everything somehow. Technological virtuality is the subset of pure virtual space that we can actually reach using available means. It is much smaller than the space of pure virtuality, but much larger and altogether different than the portion fo the virtual we can explore unassisted. Mediating between the pure and the technological is the virtual space of consciousness, the space of all the thoughts that are thinkable, at any given time or in principle, with any given set of conditions."[3]

[1] Marcos Novak, 'Liquid Architectures in Cyberspace', in *Cyberspace: First Steps* (MIT Press, 1991), 225–254.

[2] Greg Lynn, *Animate Form*, 1st ed. (Princeton Architectural Press, 1999).

[3] Marcos Novak and Alessandro Ludovico, 'Interview with Markos Novak', 2001.

3.7. Postmodern computation

The main concern of postmodernism as a philosophical paradigm was the questioning of objective truth and hence sharp, binary oppositional classifications. It postulated the reality as being subjective and subject to change, relative to time and place.

Much of postmodern thinking resulted in anti-technological trends in archiecture, criticising the concept of "deteritorialised" knowledge universal to all time and place that at the time was seemingly enforced by the use of information technology. In parallel to those trends, postmodern architecture became detached from technology focusing on cultural and symbolic aspects of its local contextuality rather than explorations of technological means allowing empowering of the architectural locality.

While main-stream architectural postmodernism focused on superficial use and adaptation of symbolic and context-specific architectural features, deconstructivist architecture took a more radical approach in rejecting the culturally determined and fixed meaning of such symbols and creation of architecture without any cultural pre-assumption. In many of his projects Peter Eisenman ignored the idea of a function as too pre-determining. Alternative approaches to architectural geometry proliferated, leading to rejection of the modular grid and exploration of new ways of addressing geometric problems. Gradually, the role of mathematical models and computers became essential to these explorations.

Contemporarily, it is clear that the information technology does not enforce absolute knowledge or culture. On the contrary, through social media, social "reteritorialisation" and a new form of the space-independent "localisation" becomes possible. With services such as Wikipedia, knowledge can be seen as something in constant motion and flux where any individual can contribute to its formation. Open to everyone means of distributing information such as Youtube videos, blogs, tweets distributed through on-line social networks provide unprecedented possibilities for localised exchange of information and knowledge, with "locality" no more being constrained by spatial proximity.

The new role of architecture in the emerging "knowledge society" is still to be defined. One approach is to see architecture as part of the complex system of continuously transforming local interactions. What remains the key role of computation in modern architecture is provision of new means for architectural activation, or in other words including events[1] in the architectural design repertoire.

3.8. Self-transforming buildings.

Seeing virtual spaces morphing and transforming on computer screens, has revived some of the interest in creation of dynamic architectural spaces in physical reality. Experiments with virtual architectural spaces were a significant inspiration for an idea of buildings able to freely transform themselves in physical space.

Already in pre-cyberspace days architects and designers were experimenting with building shapes that can change its form. In 1991 Chuck Hobberman invented the mechanism of the expanding sphere, of which some principles he applied in the Expanding Dome project of 1994. His designs illustrate the "mechanical" quality shared with many other projects that typically allow transformation from one state to another. However, it was only the generation of "cyber-architects" that have really challenged this idea.

In 1998 Kas Oosterhuis presented his first vision on the Space-Station project. In his project, the walls of a space station structure could dynamically deform to deliver various spatial qualities, provide furniture elements emerging from the walls of the space station. The

[1] Sanford Kwinter, *Architectures of Time: Toward a Theory of the Event in Modernist Culture* (The MIT Press, 2002).

innovation of this idea was a practically endless amount of spatial configurations that the space station could deliver. In 2000 Mark Goulthorpe and his design firm dECOi presented the Hyposfurface installation at the Venice Biennale, which showed a physical prototype of an architectural surface that could achieve similar complex deformations in real-time. In the meanwhile Kas Oosterhuis has continued his visionary endeavours with the WTC project for the Ground Zero site in New York; "our proposal for ground zero an open programmable architecture, self-executing emotional states and fully adjustable in shape and content (...) ready to adjust to different cultural structures and events and fit to adapt to a rich variety of use. This e-building not only reacts to different circumstances but actively proposes new configurations"[1]. Presented with it illustrations show buildings that can not only fully deform in shape and colour, but even break fixed surface topologies. An attempt to realise this vision was the Trans-Ports project (2001). Here, the concept was to create a fluid, transformable building body with a dynamic interior, augmented with interactive multimedia displays. A network of such buildings was originally envisioned to be built in several port cities (hence the name Trans-Ports). Although the project was never realised it became an incentive to build a working prototype of similar, yet simplified functionality. Called The Muscle, it became exhibited at Architectures Non-Standards exhibition held at Centre Georges Pompidou in Paris during fall 2003. This prototype created an incentive for further experimentation under the "Muscle" theme. Each consecutive semester until 2007 a group of undergraduate architecture students at Oosterhuis-directed Hyperbody group at TU Delft would design and build another transformable spatial installation[2], many of which have been supervised by the author.

Many other projects that explored the possibilities of free, dynamic transformation of architectural spaces appeared in that time. Projects such as the Leisurator[3] by a group of Architectural Association students or Topotransegrity[4] by Robert R. Neumayr explore systems that would allow physical operation of kinetically transformable architectural spaces. Other aspects of dynamic spaces are frequently touched by experimental installations from the border of architecture, art and industrial design, such as Usman Haque's "Open Burble" in which thousands of interconnected balloons, enhanced with embedded, chip-controlled multi-colour lights create a dynamic surface controlled by spectators through cables and sms messages, or "Funky Forest" by Theo Watson and Emily Gobeille, where an architectural space is dynamically augmented by interactive projections displaying a joyful ecology of computer generated plants and animals interacting with children playing in the space. Philip Beesley in the continuously developed installation "Hylozoic groud" presents a complete artificial ecosystem of plant-like spatial forms creating large and intricate, three-dimensional structures growing in response to stimuli coming from the environment[5].

[1] Oosterhuis, *Hyperbodies, Towards an E-motive Architecture*.

[2] Nimish Biloria and Kas Oosterhuis, 'Envisioning the Responsive Milieu: An Investigation into Aspects of Ambient Intelligence, Human Machine Symbiosis and Ubiquitous Computing for Developing a Generic Real-time Interactive Spatial Prototype', in *CAADRIA 2005, Proceedings of the 10th International Conference on Computer Aided Architectural Design Research in Asia* (New Delhi, 2005).

[3] Marco Pastore and Valentina Sabatelli, *Leisurator TM*, 2004.

[4] Robert R. Neumayr, 'Topotransegrity. Non-Linear Responsive Environments', in *Game Set And Match II. On Computer Games, Advanced Geometries, and Digital Technologies*, ed. Kas Oosterhuis and Lukas Feireiss (Episode Publishers, 2006), 427–437.

[5] Beesley and Armstrong, 'Soil and Protoplasm'.

Many architects also look towards the applications of technologies that have not yet left the test labs of scientists. Michael Fox and Miles Kemp list a number of possibilities of the application of Self-reconfigurable robots[1] to architecture, saying that “as architects and designers begin to adopt the technology of modular reconfigurable robotic systems, they will begin to re-envision the creation of dynamic spaces”[2]. Marcos Novak goes further in his visions, proposing “a neuroarchitecture replacing bricks and mortar with intelligent, plastic nanomaterials, keeping the central nervous system of the building informed on inner and outer influences, in precisely the same way that this occurs in the human body”[3].

3.9. Conclusion

Over the last century, the idea of creating “architecture as a machine” has evolved into a much richer vision of architecture being a dynamic system that can transform over time and through local (and subjective) interactions with its users and its environment. Recent decade has seen an explosion of experimental projects dealing with the topic of interactive and transformable spaces[4,5,6,7]. These ideas are, however, still far from mainstream architectural practice and public awareness. Many of presented concepts are independent endeavours of their authors; while relatively little consolidation or collective effort takes place in bringing those explorations together.

Clearly, much in the development of interactive architecture depends on technology. However in order to advance technological progress in this domain, a common ground needs to be established. Interactive Architecture needs a concise definition and an overall synthesising vision to drive its further development.

[1] Michihiko Koseki, Kengo Minami, and Norio Inou, ‘Cellular Robots Forming a Mechanical Structure’, in *Distributed Autonomous Robotic Systems 6*, 2007, 139–148.

[2] Fox and Kemp, *Interactive Architecture*.

[3] Thomas Markussen and Thomas Birch, ‘Minding Houses’, In *Intelligent agent*, Vol. 5, No. 2, 2006.

[4] Bullivant, *4dspace*.

[5] Lucy Bullivant, *Responsive Environments: Architecture, Art and Design* (Victoria and Albert Museum, 2006).

[6] Lucy Bullivant, *4dsocial: Interactive Design Environments* (Wiley, 2007).

[7] Aaron Betsky et al., *Disappearing Architecture: From Real to Virtual to Quantum*, 1st ed. (Birkhäuser Basel, 2005).

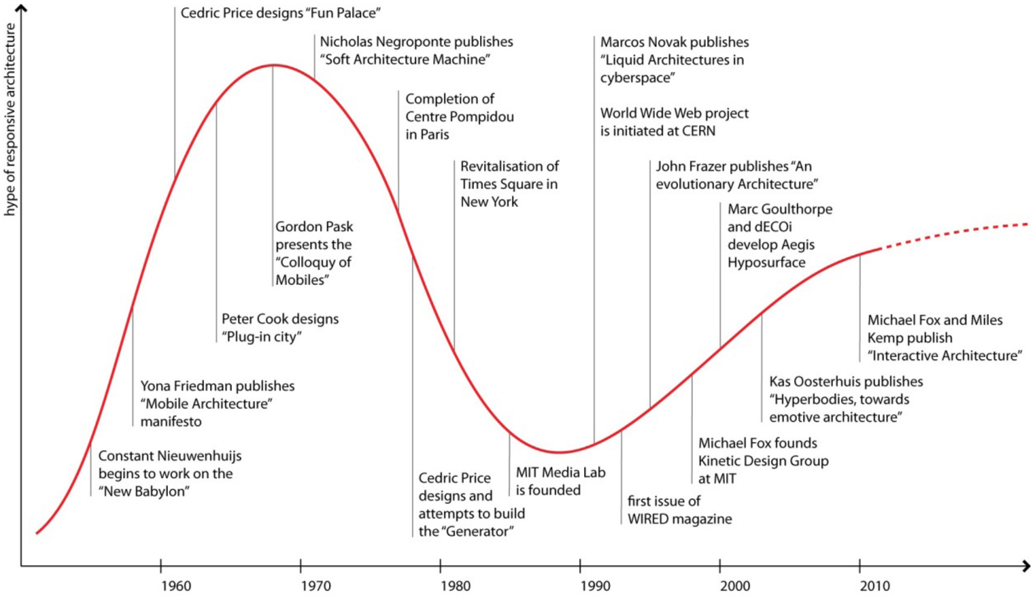


Fig.9. Gartner's hype cycle[1] adapted to represent the development of responsive architecture.

4. Interactive architecture

In the discourse about "active" architecture, many notions have appeared in the course of the strive to provide the most appropriate label for the new kind of buildings and spaces that can change and communicate with their users. The term "automated architecture" would be the kind of architecture where "building automation" is paramount. Automated architecture may be just like the traditional one, yet in which repetitive actions of users are facilitated by the building. Escalators, elevators and tea serving robots take the burden of even having to think about daily chores off the inhabitants of automated buildings. The "smart architecture" with sub-domains of "smart buildings" and "smart homes" is the continuation of the automated architecture line of thinking. Its premise is that in order for architecture to help its inhabitants certain level of "intelligence" (hence also the terms "intelligent building", "intelligent house") is needed to understand user intentions and to synchronise often conflicting activities. The idea of "reactive architecture" underlines the way of operation, where every action of an architectural construct is performed as a direct reaction to user or environment, presumably following a pre-programmed pattern of behaviour. The concept of "responsive architecture" enhances this idea to emphasise the occurrence of "feedback loops", where there action-reaction chain has no end and resembles a "dialogue". The concept of "transactive architecture" underlines the importance of an occurring transaction in such communication process between architecture and its users, as well as mutual transformation incurred by such transaction.

[1] Jackie Fenn and Mark Raskino, *Mastering the Hype Cycle: How to Choose the Right Innovation at the Right Time* (Harvard Business School Press, 2008).

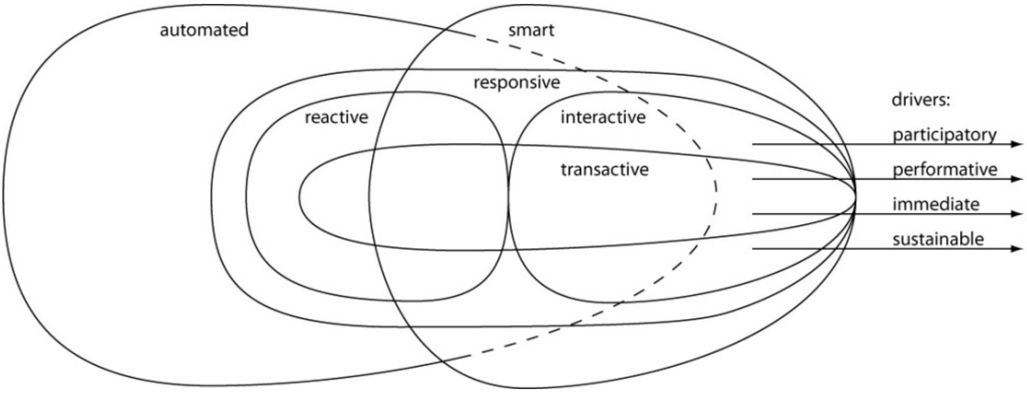


Fig.10. Sub-domains of active/dynamic architecture and their common drivers

On the other hand, there are other ideas, which although imply architecture being active, don't explicitly point to any sort of a communication process through which that could be achieved. Instead they function as "drivers" of sort, providing explicit purposes, phenomena for which dynamic architecture is a vehicle. The term "performative architecture" implies architecture designed with conscious understanding of the way it is going to perform. In its original understanding such performance can was mostly understood in a structural way; however it can equally apply to energy use performance or any other measurable quality of architectural operation. "Participatory architecture" focuses on the role of users from the early point of the development of architectural design, but also in redesigns and reconfigurations during the building operation. The idea of "immediate architecture"[1] means architecture that can be designed, (re)built or modified and used at the same time. Arguably, all these ideas can be seen as qualities of architecture more than specific domains. Surprisingly, the idea of "sustainable architecture" can be seen as a similar quality of active architecture, underlining its ability to self-sustain through its own actions.

Other concepts include "kinetic architecture", where the emphasis is put on architectural constructs being able to move and physically transform themselves, or "robotecture" where architectural spaces are envisioned to encompass robotics. In both cases these terms refer to means of achieving certain qualities or behaviours rather than justifying their "whys? "or "hows?".

The notion of "interactive architecture" is not attempting to compete with all listed domains or replace them. On the contrary, it attempts to synthesize all listed concepts and provides common ground and a more explicit focal point for all of them. At the same time, it clearly indicates a direction of future development, towards enrichment of reciprocal relations between users and architecture, or in other words an increase of their complexity and decrease in their complicatedness. Following points attempt to establish a universal definition of interactive architecture.

4.1. Interactivity

Clear definition of the notion of interactivity is fundamental for further clarification of the concept of interactive architecture. The term *interactive* is used widely in contemporary culture, especially in relation to new media and computer systems. However, in common

[1] Christian Friedrich, 'Immediate Architecture', in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012), 225–260.

language this notion often appears ambiguous and is frequently overused to denote any product or service that employs computer technology. Therefore to establish grounds for further argumentation interactivity requires a structured and cohesive definition.

Interactivity is normally understood as a process, constituting of a chain of inter-related actions of two or more interacting subjects. "(Interactivity is) a bidirectional conduit. It's a response. Interaction is a relationship. It's mutually executed change"[1]. Such relation between interacting subjects is inherently reciprocal and interacting subjects always become in some way transformed or otherwise changed in the course of interaction.

From another perspective, this definition may be further extended by giving interactivity a variable degree. Some interactions intuitively seem "more interactive" than others. Such degree of interaction is often proportional to the amount of exchanged information and/or the amount of change this exchange invokes on interacting subjects (yet exceptions may be easily found). In order to use the term *interactivity* as a variable in the scientific context, a more precise definition has to be introduced.

One of the seminal interpretations of the notion of interactivity belongs to Sheizaf Refaeli. According to his definition, interactivity is a variable being an "expression of the extent that in a given series of communication exchange, any third (or later) transmission (or message) is related to the degree to which previous exchanges referred to even earlier transmissions"[2]. As its author further argues, this definition also implies that interactive communication has to be distinguished from its two-way and reactive variants.

This description, seemingly complex at first, comes down to a very practical classification of communication types that may occur between two or more subjects. When messages being exchanged are sent in two ways, but are not related to each other, *two-way communication* occurs. If each of the messages sent in a process relates to a previously received message only, we are dealing with reactive communication. If this relation is more complex and each message being sent relates to more than one message received in the past by the answering subject, we are dealing with interactive communication.

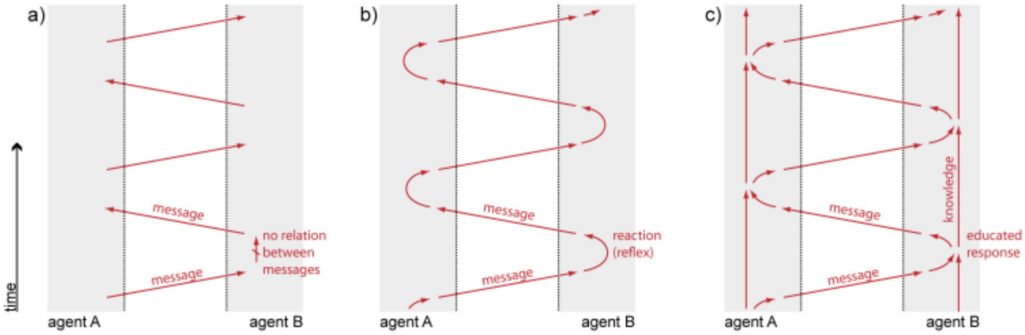


Fig.11. information flow in a) two-way, b) reactive and c) interactive communication, based on S. Rafaeli

In this way, interactivity can be defined as a measurable variable, which occurs (is greater than 0) only when a communicating subject has ability to relate its response to information received before the one that triggers its immediate reply. Value of interactivity increases with the number of earlier received messages used to formulate response at a given moment.

[1] Mark Stephen Meadows, *Pause & Effect: The Art of Interactive Narrative* (New Riders Press, 2002).

[2] Sheizaf Rafaeli, 'Interactivity: From New Media to Communication', in *Advancing Communication Science: Merging Mass and Interpersonal Processes*, ed. R. P. Hawkins, J. M. Wiemann, and S. Pingree, vol. 16, Sage Annual Review of Communication Research (Sage Pubns, 1988), 110–134.

Following this definition, interacting subject's ability to store and process information is a prerequisite for interactive communication to occur, accounting for its reciprocity and resulting in its increased agency.

Rafaeli himself revised his own definition to be: "the extent to which messages in a sequence relate to each other, and especially the extent to which later messages recount the relatedness of earlier messages"[1]. This and other definitions became gradually more biased in their target applicability to human-computer interaction[2]. On the other hand, many other definitions that are applicable exclusively to understanding of interactivity in specific fields of hard and soft science[3] also exist. For the clarity of argumentation, the first definition of Rafaeli will be used throughout this research as the most generic and yet unambiguous one of known definitions. However, other definitions, such as this of Gordon Pask, will be referenced, yet always presented in logical relation of Rafaeli's.

The notion of *interactivity* also has to be clearly distinguished from the term interaction. Interactivity can be seen as a process of multiple interactions, where interaction may be any action occurring when two or more subjects affect each other, therefore possibly being part of interactive, reactive or two-way communication process. Consequently, an expression "non-interactive interaction" may intuitively appear as an oxymoron, but in fact is a just term referring to interactions in two-way and reactive communication processes[4].

4.2. Interactive architecture

Following the earlier presented definition of interactivity; the notion of interactive architecture can be provisionally defined as buildings and built spaces capable of sustaining active dialogue with their users and environment. A word *dialogue*, used here metaphorically, refers to the conversation theory of Gordon Pask[5] and the idea further promoted by, among others, Usman Haque saying that "if one wants occupants of a building to have the sensation of agency and of contributing to the organisation of a building, then the most stimulating and potentially productive situation would be a system in which people build up their spaces through 'conversations' with the environment, where the history of interactions builds new possibilities for sharing goals and sharing outcomes."[6]

Regardless of the attractiveness coming from its novelty, there are multiple other, practical reasons for architecture to be interactive. Interactivity as an architectural feature can provide a viable solution to the growing need for adaptive and customizable spatial qualities in buildings. At the same time interactive architecture is also a compelling alternative to building automation, having the potential to radically change the way in which buildings perform, are used and maintained. Building automation undeniably led to creation of architectural spaces which are active in many of their aspects. However, using Rafaeli's terminology, automated spaces perform in a purely reactive manner. Only consequent replacement of linear logics that guide building behaviours with ability to autonomously reason and learn may result

[1] Sheizaf Rafaeli and Fay Sudweeks, 'Networked Interactivity', *Journal of Computer-Mediated Communication* 4, no. 4 (1997).

[2] Edward J. Downes and Sally J. McMillian, 'Defining Interactivity: A Qualitative Identification of Key Dimensions', *New Media Society* 2, no. 2 (June 1, 2000): 157–179.

[3] J. F. Jensen, 'Interactivity. Tracking a New Concept. In Media and Communication Studies', *Nordicom Review 1/1998. Special Issue: The XIII Nordic Conference on Mass Communication Research* 1, no. 19 (1998).

[4] Paul Ferber, Franz Foltz, and Rudy Pugliese, 'Interactivity Versus Interaction: What Really Matters for State Legislature Web Sites?', *Bulletin of Science Technology Society* 25, no. 5 (October 1, 2005): 402–411,

[5] Gordon Pask, *Conversation Theory, Applications in Education and Epistemology*, 1976.

[6] Usman Haque, 'Architecture, Interaction, Systems', *Arquitetura & Urbanismo*, 2006.

in their interactivity. This implies creation of architectural spaces being able to maintain a continuous dialogue not only with their users but also between each other, instead of reacting to pre-programmed conditions. In other words, interactive architectural spaces would not only respond to predefined demands, but actively engage themselves in all kinds of activities taking place within their context.

The contemporary understanding of interactive architecture underlines the inherent complexity present in interactive spatial systems, which need to consist of many components, have to deal with many users and don't have easily definable performance criteria. Citing Branko Kolarevic; "process-based (...) architecture in itself has the characteristics of an ecological system that emulates nature instead of protecting it and therefore engages in an enduring fusion of nature and culture".[1]

There have been many attempts to clarify the notion of interactive architecture in a concise description. "Interative Architecture (iA) is NOT simply architecture that is responsive or adaptive. On the contrary iA is based on the concept of bi-directional communication (...) it is first defined as the art of building relationships between built components and second, as building relations between people and built components"[2] , "It [interactivity in architecture] is about designing tools that people themselves may use to construct (in the widest sense) their environments and thus to build their own sense of agency. It is about developing ways to make people themselves more engaged with, and ultimately responsible for, the spaces that they inhabit. It is about investing the production of architecture with the poetries of its inhabitants."[3]," Interactive Architecture [, for me] is about the potential for digital systems to make decisions about our living environment and then influence that environment."[4], "the current landscape of interactive space is built upon the convergence of embedded computation (intelligence) and a physical counterpart (kinetics) that satisfies adaptation within the contextual framework of human and environmental interaction (...) motivation to make these systems is found in the desire to create spaces and objects that can meet changing needs with respect to evolving individual, social, and environmental demands."[5] "It is the [architectural] form that is no longer stable, that is ready to accept change. Its temporary state is determined by the circumstances of the moment on the basis of an activated process and in-built intelligence and potential for change. Not product architecture then, but a process-based architecture whose form is defined by its users' dynamic behaviour and changing demands and by the changing external and internal conditions; an architecture that itself has the characteristics of an ecological system, that emulates nature instead of protecting it and therefore engages in a enduring fusion of nature and culture."[6] There have been many other attempts that approach the definition of iA and its sister concepts (such as responsive architecture or smart environments). To avoid ambiguity, author's own synthetic definition will be used canonically throughout presented research: "interactive architecture is architecture that exhibits autonomous behaviour, in which that behaviour evolves through interactions with its users and environment".

Interactive architecture does not yet comprise fully applied precedents. What's more, there are no comprehensive models or frameworks, neither technological nor cultural, in which interactive architecture could yet effectively operate (the irrelevance of contemporary building automation systems will be discussed further). Therefore, taking the novelty of this

[1] Branko Kolarevic, 'Exploring Architecture of Change', 2009, .in *reForm*, ACADIA 2009 Annual Conference, School of the Art Institute Chicago

[2] Kas Oosterhuis and Xin Xia, *iA #1* (episode publishers, 2007).

[3] Haque, 'Architecture, Interaction, Systems'.

[4] Oosterhuis and Xia, *iA #1*.

[5] Fox and Kemp, *Interactive Architecture*.

[6] Kolarevic, 'Exploring Architecture of Change'.

field into account, it is essential to justify the need for further investigations into this area in form of a brief overview of arguments and counter arguments for and against potential development of interactive architecture.

Implementation of interactivity in architecture is not inherently related to any particular functional quality, applied technology or method. Its fundamental nature lies in introducing users as dynamic actors into the core of building design and performance processes. It also means that buildings or building components have to be considered as dynamic actors, the behaviour of which cannot be predetermined. This new kind of architecture requires revision of ways in which architectural design and all related domains are dealt with. Any associated technological developments should be seen as consequences of this shift in building nature, rather than enforcing the common preconception of technology driving the development of architectural interactivity. As prof. Antonino Saggio summarises: “Firstly, interactivity is now the catalysing element of architectural research and development activity because it is within this that the contemporary communication system, based on the possibility of creating metaphors and so of firstly navigating and then building hypertextual systems, resides. Secondly, interactivity places at its centre the subject (variability, reconfigurability, personalisation) instead of the absolute nature of the object (serialisation, standardisation, duplication). Thirdly, interactivity incorporates the fundamental feature of computer systems, namely the possibility of creating interconnected, changeable models of information that can be constantly reconfigured. And lastly, interactivity plays, in structural terms, with time, and indicates an idea of continuous ‘spatial reconfiguration’ that changes the borders of both time and space that until now have been consolidated.”[1]

4.3. Architectural adaptation through interactivity

“Evolution now must include evolving environments which evolve man, so that he in turn can evolve more propitious environments in an ever quickening cycle. To stabilize the capacity we need to characterize this evolutionary dialogue. This characterization is increasingly being seen as the unsolved problem of our time. It is familiar to designers and architects in the student’s question: ‘How do you design a house which will grow to meet the changes in the family that the house itself will produce?’”[2]

The challenge set for interactive architecture is to allow high frequency of adaptation, comparable to that which occurs in automated systems, but at the same time to allow it to adapt to unforeseen conditions, such as in the case of spatial reconfigurations. If we apply the Rafaeli’s definition of interactivity to categories listed earlier, we may realize that within a certain scope of time all of given examples can be actually defined as interactive.

If we start with reconfigurability, we may say that each reconfiguration is an act of exchanging information between the building or built space and its users seen as a group. Users communicate their needs by altering the space and the space responds by allowing certain spatial performance. It relates to all previous acts of reconfiguration, because information in this case is inherently stored in the form of the building. However, this point of view is only possible if we accelerate common perception of time beyond its usual scope of concern. The value of interactivity would be low in this case, not because the exchanged information doesn’t relate to many earlier exchanges, but because there are normally at most only several acts of such “communication” taking place throughout the building lifecycle.

[1] Antonino Saggio, ‘Interactivity at the Centre of Avant-Garde Architectural Research’, *Architectural Design* 75, no. 1 (2005): 23–29.

[2] Brodey, ‘The Design of Intelligent Environments: Soft Architecture’.

On the other extreme, in automated buildings communication between users and dynamic installations occurs at very high frequency, often many times per second. However, since the system only responds in a pre-programmed manner, there is none or little interactivity taking place.

In between there are portable or flexible solutions that fill the spectrum of possibilities by different balances between frequency of building-user communication and openness to unplanned kinds of adaptability, however in all these examples there seems to be sacrifice of one of these features by cost on the other.

There are also hybrid solutions. An automated building can be reconfigured, together with its automated installations, it may contain flexible and portable components, however even if this is the case, there is usually no connection between these processes, mostly due to different temporal and spatial scales on which they operate.

Until this point the notion of scale has not been explicitly addressed. It is however of very high relevance, mostly due to hierarchical manner in which we traditionally deal with built spaces. Quoting John Habraken[1], "rooms make the house and houses make the neighbourhood and neighbourhoods the town". This scale dependent division of design tasks led to separation of professions that deal with spatial design and let to linear approach where first the urban planner designs a city plan, then an architect designs a building, after that engineers design structure and installations and eventually interior designer the finishing and furniture, without any feedback mechanisms. Because of this, even if the furniture in the house can be seen as portable, flexible or automated, the house itself (on the scale of architectural design) at the same time may not have any of these features. Therefore, interactivity in architecture requires not only connectivity between different installations that are scale-wise on the same level, but also bi-directional communication between entities on various scales. In other words, not only urban scale has to inform architecture and architecture has to inform installations and interior design, but also interior has to inform installations, architecture which in turn has to inform urban design. All components that constitute special environments, on similar and varying scales have to operate in continuous feedback loops.

Two main approaches can be taken to achieve this integration. One approach would be to integrate everything in a deterministic system, being either centralized, decentralized or distributed. The other approach, further pursued in this research is to develop such system as non-deterministic, allowing for its emergent properties. The specific approach explored in the following chapters investigates creating such system as made up of autonomous adaptive agents.

Examples of truly interactive spaces are scarce and never comprehensive, usually taking form of experiments connecting art, architecture and technology. Various paths can be taken to approach interactive architecture and any of the four categories of existing adaptive spaces can be taken as point of departure. Ultimately architectural interactivity can be only achieved as synergy of these approaches.

Instant city project of Archigram is one of the first design concepts that can be described as interactive architecture. Although never built, or eve technologically investigated, it provides a compelling vision of an architectural space being an event in itself and dynamically moving to different locations, engaging itself with new participants and contexts.

Also unbuilt, two projects of Cedric Price; the Fun Palace from 1961 and the Generator from 1979, envisioned the dynamic spatial environment. What is prominent in both, the projects of Archigram and Price is the combination of automation and reconfigurability to provide ability

[1] John Habraken, 'The Control of Complexity' 4, no. 2, Representations and Realities (July 1, 1987): 3–15.

of architectural constructs to be easily reshaped in unpredictable ways. What's more, in both cases no centralised control mechanisms are there to guide those spatial behaviours. The control is handled in an entirely distributed manner.

Despite their undisputed potential, none of these early visions of interactive spaces were realized. One of the first projects of similar kind that ultimately became built is the Media House by MIT Media Lab, Fundació Politècnica de Catalunya and Metapolis. The leading concept in this project is the employment of Internet0[1]; "internet of things" in combination with a freely reconfigurable building structure, which at the same time is also a network installation for the building. In this installation built in 2001, all sensors and actuators could be directly plugged to the building and their behaviour defined over the internet or manually by building's users. Even though it was not explicitly demonstrated, certain degree of agency was present in the system. Although the installation as such was purely responsive many users controlled its operation simultaneously, certain degree of interactivity was purely reactive in its behaviour, the system seen as a whole with multiple users and installation components may be described as interactive.

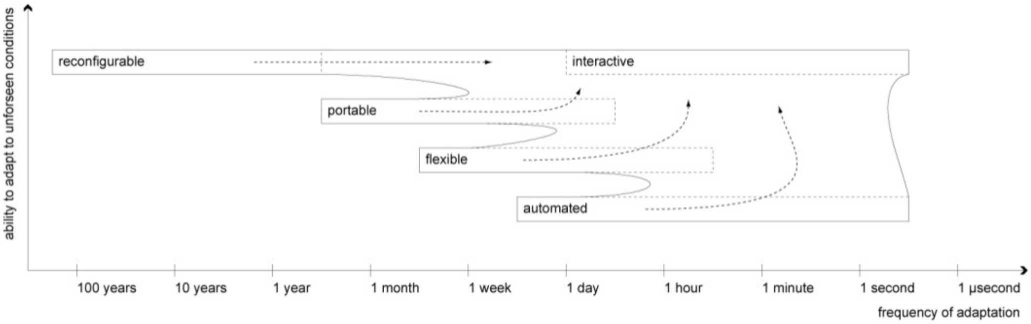


Fig.12. Convergence of existing adaptive architectural types towards interactive architecture.

Another example is the Reconfigurable House 2.0 by Usman Haque and Adam Somlai-Fischer. It is based on a similar concept as the Media House; however the main difference is the use of commonly available, hacked low-tech devices as constituent components of the installation. "The many sensors and actuators of Reconfigurable House can be reconnected endlessly as people change their minds so that the House can take on completely new behaviours (...) and if the House is left alone for too long, it gets bored, daydreams and reconfigures itself"[2]. The reconfigurability of the installation using simple, commonly available cheap devices makes it possible for reconfigurations to happen very frequently, on the other hand the open nature of the system and relative ease by which it can be reprogrammed make its automated features easily adaptable to unpredictable conditions. The project's functionality unfortunately doesn't go beyond this of an art installation, yet it can be seen as a first step towards achieving complete spatial interactivity.

[1] N Gershenfeld and D. Cohen, 'Internet 0: Interdevice Internetworking - End-to-End Modulation for Embedded Networks', *Circuits and Devices Magazine, IEEE* 22, no. 5 (2006): 48–55.

[2] Adam Somlai-Fsicher and Usman Haque, 'Reconfigurable House', accessed March 23, 2009, http://house.propositions.org.uk/.

5. Life in iA

The concept of rapid and non-predetermined adaptation of space through interactions involves a fundamental transformation of the way in which architecture functions in human society. Rather than living our lives in architecture, we are to live *with architecture. The rudimentary question to be answered is just how interactive architecture can operate in the society from the pragmatic perspective of its users?*

5.1. Envisioning iA

There can be numerous ways for creation of iA to be attained technologically. Yet, technology is not the key factor in determining the fundamental nature of spatial interactions. Technology may merely provide means to an end, which is directly determined by the needs and activities of inhabitants and how they can take advantage of dynamically changing and evolving architectural spaces through interactions with it.

In order to understand the specific nature and complexity of possible interactions between people, architecture and environment, let's consider existence of a hypothetical architectural environment made of unspecified "programmable matter"[1] or, more theoretically, Marcos Novak's "alloatoms"[2]. Let this imaginary matter be able to sense, interpret received signals and adapt to such sensed conditions by freely changing its own shape, texture, colour, transparency and other material properties and by transforming energy drawn from various sources into light, heat or sound. Let this matter have also an ability to alter its own behaviour governing its adaptations, by learning from received feedback. Let's then try to analyse different scenarios of interaction involving such "living" architecture, starting from the most local and building up to large scale phenomena.

5.2. Local spatial adaptations through interactions with individuals

As discussed in section 1. of this chapter, there exists a reciprocal relationship between activities of people and the architectural context of these activities. It can be determined from particular actions of an individual which spatial conditions and facilities are required to facilitate performed activity. Based on a given state and behaviour of an individual, along with its context and previously observed patterns, individual's activities can also be to a certain extent predicted, allowing for anticipatory adaptations facilitating or enabling execution of user activities that did not yet occur, but which are very likely to take place. On the other hand, existence of particular spatial conditions and facilities can influence one's needs and activities by creation of specific affordances and other means of pro-active suggestion of use.

This relationship can be translated to and infinite number of scenarios of possible one-to-one architecture-inhabitant interactions. In such scenarios, responses of architecture can encourage an already occurring or expected activity of a person, or discourage it. The strength of encouragement or discouragement can vary from subtle suggestion to enforcement. Multiple activities can be differently encouraged or discouraged at the same time. In this way iA promises an environment, which with a very high efficiency and speed "understands"

[1] S.C. Goldstein, J.D. Campbell, and T.C. Mowry, 'Programmable Matter', *Computer* 38, no. 6 (May 2005): 99 – 101.

[2] Marcos Novak, 'Transvergence: Finite and Infinite Minds', in *Game Set and Match II: The Architecture Co-laboratory on Computer Games, Advanced Geometries, and Digital Technologies*, ed. Kas Oosterhuis and Lukas Feireiss (Episode Publishers Rotterdam, 2006).

the needs of its users and instantly caters to them. Such system could also anticipate needs of its users based on their prior activities and in this way increasing the “efficiency” of its behaviour even further. On the other hand, many the performed adaptations of the system would also pro-actively influence the needs of systems’ inhabitants, tightening the reciprocal interdependence of spatial needs and affordances, changing user beliefs and cultural patterns.

The simple example of a typical human activity is resting. Every human being needs a certain amount of rest in a day. Lack of rest can have negative or even fatal influence on one’s well-being. On the other hand, excessive rest can also lead to decrease of physical and psychological well-being. When a person sits or lies down and significantly reduces its movements, it can be easily interpreted as an act of resting. Spatial affordance for such activity may include provision of furniture or other spatial arrangement allowing appropriate support for resting human body, dimmed lights, and lack of noise. However, in many cases it is in person’s interest not to rest. For example, there may be other, more urgent activities to perform, one has rested for too long and needs more physical movement to maintain good health, or there is an emergency situation requiring evacuation. In such cases it is necessary that spatial conditions motivate the resting person to change the activity. Therefore, there are three aspects in spatial response needs to be “smart”. It needs to properly deduct the nature of inhabitant’s behaviour from sensed conditions, e.g. not to confuse resting with cooking. Subsequently it needs to determine if that activity should be enforced or opposed. Eventually, it has to change spatial conditions in a way that will lead to such enforcement or opposition.

Even if the intelligence of the hypothetical “smart” architectural system surpassed human intelligence, there would still be high risk of mistakes on any of the three steps in the logics behind considered adaptation process, namely: a) interpretation of acquired data b) intention of adaptation c) execution. A wrong assumption in any of these steps may have devastating consequence. Using the example of resting, a misinterpretation of data or wrongly made decision could lead, for example, to fatal sleep deprivation of an inhabitant or failure of his or her evacuation in case of an emergency.

What’s more, based on a small flaw in logical reasoning, an unconstrained learning system could potentially deduct that harming its inhabitants is of benefit from the point of view of the set criteria of the adaptive process. Such possibility causes a lot of concern. It is a subset of problems generally encountered in respect to creation of autonomous robots, where an interactive building can be considered to be a specific kind of a robot. This topic has been widely dealt with in popular science and science fiction. The term robot originates from a play “R.U.R. (Rossum’s Universal Robots)” by Karel Čapek, written in 1920 where “robots” begin to disobey humans and eventually eradicate them. These concerns were greatly elaborated further in the literary works of Isaac Asimov, where many subtleties and ethical concerns related to development of robots are featured. Asimov’s work became a foundation not only for much of the science fiction work related to robots, but also strongly influences the developments in engineering of robotics. Among robotic visions involving the “architectural” aspects of robots, the film “Space Odyssey: 2001” by Stanley Kubrick is one of the most recognised. In this film the spaceship on which the action is set is fully controlled by a supercomputer HAL. In the course of action HAL decides that the astronauts are a threat to mission objectives and decides to eliminate them.

In responses to such concerns, Asimov formulated the "laws of robotics", which have gradually crystallised throughout his works[1] (0. A robot may not harm humanity, or, by inaction, allow humanity to come to harm 1. A robot may not injure a human being or, through inaction, allow a human being to come to harm. 2. A robot must obey any orders given to it by human beings, except where such orders would conflict with the First Law. 3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.). Asimov's "laws" may be too ambiguous to apply them directly to a robotic system2; however they could potentially serve as a basis for a more specific, fixed rules and constraints for operation of iA, being considered as a robotic system of sorts. An example of such rules could be a) well-being of inhabitants is the highest priority of an iA system and it performance is evaluated by the ability to cater to that well-being b) inhabitants need to be able to provide unambiguous feedback at any moment, and/or override the autonomous behaviour when necessary) c) decisions taken by the system need to be continuously re-evaluated by its inhabitants d) behaviours that can be potentially life-threatening need to be ruled out by fixed procedures in the system.

5.3. Multi-user adaptations

In architectural setting there is rarely only one occupant. Often there are none, in many cases spaces have multiple users. Several users in one space can share an activity, yet often they can be engaged in different activities which require different affordances and hence can generate conflicts.

The role of spatial adaptation is thus not only to cater to individual needs of its users. It is also to resolve conflicts between those users. Ideally such conflicts can be solved in a win-win manner, by finding spatial solutions that satisfy all. In many cases, however, win-win solutions may not be possible. For this certain activities or certain users need to be prioritised over others. Prioritised users may be the owners or assigned care-takers of the space. Priority of activities can be decided upon various criteria, such as estimated increase of well-being of people involved in an activity, the urgency of an activity, the purpose of an activity, involved cost, or minimising dissatisfaction of people whose activity cannot be supported. These criteria can also change over time, and can be e.g. directly influenced by prioritised users.

The simplest example of a non-conflicting multi-user activity is a social discussion, where multiple participants are required for the activity to at all take place. Similarly, during another activity such as a lecture, there needs to be at least one presenter and at least one spectator. Although individual activities of the presenter and spectator differ (one is presenting, the other one is observing), they form one activity pattern. On the other hand, two activities of listening to a lecture and having a social meeting are conflicting. Persons listening to a lecture can be disturbed by noise produced by those having a social discussion. In this situation a remedy is acoustic separation of the two groups, which can be introduced as a spatial adaptation. However, based on assigned priority to those activities, one group or the other could be motivated to terminate their activity or move to another location which affords their activity better.

[1] Roger Clarke, 'Asimov's Laws of Robotics: Implications for Information Technology-Part I', *Computer* 26 (1993): 53–61.

[2] Roger Clarke, 'Asimov's Laws of Robotics: Implications for Information Technology-Part II', *Computer* 27 (1994): 57, doi:10.1109/2.248881.Dec 1993, p53-61.

5.4. Multi-space adaptations

Conflicting activities lead to creation of diverse architectural spaces. Although each space may have a different function (permanently or temporarily) and thus cater to different activities, there can be a strong correlation between architectural spaces, as activities of people are interconnected. For example, any activity of people requires proximity of sanitary facilities. Many other activities are mutually enforcing, such as a group of people coming together to see a lecture is likely to engage itself in social discussions. In this way, spaces with distinct functions form clusters of higher level functionality (traditionally labelled as e.g. “shopping mall” or “airport”).

Architectural functions are traditionally seen as permanent, which means they only cater to a pre-defined set of activities. However through hypothetically unconstrained dynamic adaptations, the range of activities to which a given space can cater can be infinitely extended. What’s more, certain spaces may only need to be distinctly formed when explicitly needed.

There are two main types of conflicts between distinct architectural places. The amount of space available for all functions required in a given context may be limited due to the external conditions, or due to the available amount of material and energy. In that case, increase in size of one space requires the reduction of other. The other type of conflict stems from the activities performed within distinct spaces, which may be affecting each other’s physical qualities (e.g. shading) or culturally conflicting (e.g. a brothel next to a church). Dynamic adaptation of spaces can help to minimise such conflicts. The first conflict can be resolved by dynamically changing the spatial geometry of given architectural spaces in order to minimise the amount of unused space at a given time. The second conflict can be solved by continuous monitoring of all relations and dependencies between activities performed in involved spaces and continuous adaptation of spatial organisation with respect to those relations and dependencies.

In this way an interactive architectural system can be envisioned as a network of distinctly functioning spaces, which can both internally adapt to particular activities, as well as adapt to each other. Such adaptations can include changes in the global layout and topology, as well as on going disappearance, re-appearance and creation of new spaces.

The same model of multi-space adaptation can be applied not only to a building, but also to open public spaces. Eventually, by pushing system boundary even further, it can expand to involve urban-scale systems where multiple buildings adapt in an interconnected manner, in synchrony with continuous adaptation of public spaces, transportation networks and other spatial transformations.

5.5. Adaptation to external factors

System boundary cannot be pushed indefinitely. External factors can be defined as all input to the iA system originating from the outside of its assumed boundary. An external factor can be either of artificial or natural origin. Similarly to changes in internal factors predictions of external factors can also be made within the system. It can also be estimated if any specific change in the system itself will affect these external factors in the future. The distinction between internal and external factors is mostly one of an agreement. In theory any agent causing an external factor could be internalised in the system. However, too great extension of boundary becomes impractical in case of artificial factors and goes beyond the concern of architecture in case of natural factors. To provide an example, a boundary of the iA system can be aligned with the boundary of a building, where everything affecting the building and its inhabitants, but located outside of the building can be considered as an external factor. However, it is essential to note that his boundary can be defined in different ways. For example it can correspond to a physical scale of a city or on apartment. It can also correspond to a specific community of people and the environment they inhabit, without a coherent spatial boundary.

Most prominently changing external influence on a building comes from weather. A dynamic building system can thus continuously change its internal organisation and encourage activities that are possible due to current given weather conditions. Open terraces with maximised sun exposure on a sunny spring day that become enclosed and shrink during winter and at night are just one possible example. An artificial external factor, such as a firework show (or any other outdoor cultural event) is one of many examples when adaptation to weather and night would be overrun to direct the said terraces to provide best firework viewing experience to their occupants. The number of other external factors to which a building can adapt is endless. Such factors can range from economic or political to natural disasters. In many cases an ability to adapt can not only improve well-being of building inhabitants but also prevent building's destruction.

5.6. Architectural: behaviour, learning, development and evolution

Hypothetically envisioned interactive architecture exhibits traits that are new to architecture. Firstly, interactive architecture exhibits *behaviour*. Its behaviour governs the manner in which iA adapts to individual needs of its users, how it resolves conflicts between them, how it deals with conflicts and opportunities coming from other related spaces and how it responds to external factors in its environment.

The behaviour of an iA system can be improved through learning. Learning is an essential quality of an interactive system, as otherwise it is only capable of providing automated; predefined responses constrained to predefined behaviour scenarios.

Through continuous spatial adaptations driven by behaviour and learning from past experience, an iA system also develops over time. This development may have a form of physical change akin to growth. It may also result in development of other spatial qualities ranging from material properties to spatial organisation. Successful development of an iA system means increase in its multi-objective performance.

In the process of adjusting the behaviour and development of an iA system successful solution are encouraged and unsuccessful ones are removed from the system. New solutions can also be gradually introduced to improve different aspects of system operation. In this way each particular system evolves over time. Exchange with other systems is possible, guaranteeing global evolution of iA, perhaps branching into more distinct "species" in the future.

6. Conclusions

The chapter has set off with the premise to investigate the nature of adaptations and interactions in architectural systems and to draw conclusions for the foundations of the iA development framework in respect to the scrutinised concepts of adaptation, interactivity and interactive architecture.

6.1. Summary

- Every adaptation in architecture can be studied as a form of interaction between two entities.
- In such process the commonly perceived interacting entities are: a) building, b) inhabitant; as an individual or a social group c) stakeholders and experts d) natural environment, e) artificial context.
- Traditionally humans play the role of active agents in adaptive architectural systems, while other entities are considered to be passive, or alternatively as mediums for interactions between humans.
- Typically increase in building automation reduces building's ability to unforeseen adaptations, while increasing the spread of predictable adaptations. The challenge for interactive architecture is to allow high speed of adaptation, while not hindering the adaptation of architecture to unforeseen circumstances.
- For reasons stemming from economic, cultural and climate transformations, interactions in the built environment are becoming less cyclical, thus less predictable, mandating increasing demand for adaptation and flexibility of buildings.

6.2. Expectations

Most generally, interactive architecture is expected to deliver fundamental increase in broadly understood "architectural performance", which can be defined as human habitat's ability to autonomously cater to the needs of its inhabitants. One aspect of iA providing such increased performance is continuous self-improvement of the built environment based on constantly gathered feedback. The other aspect is continuous adaptation of the built environment to changes in human needs and changes in the external environment.

Conceptual visions of self-adaptive architectural environments bring many promises. Improved catering to human well-being can be achieved by increased temporal specialisation of building spaces. In other words, it means providing affordances and facilities exactly as required in given conditions, but only for the duration when they are explicitly needed. In this way iA has the potential to deliver unprecedented multi-objective optimisation of continuously re-created spatial organisation. It may also lead to significant reduction of the use of resources such as material, energy, time and space. What's more, it can guarantee sustainability of architectural systems, through continuous feedback between those systems, their users and the natural environment.

Elimination of redundancy in iA, contrary to what occurred in modernist architecture, is not to be paired with diminishment of architectural aesthetics and expression, or any other qualities of architecture commonly considered as subjective. In iA systems personal preferences of users directly inform architecture, leading to customisability of any of its aspects that are not otherwise constrained by higher-priority factors. What's more, traditional "beauty" of architecture is no longer to be confined in a static form, but it is to involve the behaviour of architecture, opening a whole new area of possibilities for architectural expression.

It is not to be claimed that all above qualities are to be inherent in any iA system. The difference between iA and traditional architecture is that iA can gradually develop and evolve its behaviour, continuously improving its ways and means of spatial adaptations. Thus, although many of the listed features are speculative, an appropriately created iA system should be able to evolve into a system that exhibits all expected properties.

6.3. Risks

In order to move beyond pure speculation, the actual development of interactive architecture requires new approaches to architectural design and engineering, production and construction, as well as operation and use of architecture. Not only roles of professionals responsible for creation of architecture need to be reinvented for iA. Also the culturally and societal role of architecture is bound be radically transformed.

Traditional architecture is profoundly connected to human culture. It is unknown what would be the long term impact of interactive, autonomously adapting buildings on human psychology and in consequence on society and culture. Currently observed reactions to prototypical interactive architectural installations are a mixture of technological excitement and anxiety, which can be fully attributed to the novelty factor and will not play a long-term effect.

It can only be speculated that an environment which continuously transforms may involve certain unwanted cultural and psychological phenomena. One possibility is that continuously transforming architectural spaces may deprive humans of the sense of identity of place and may trigger feelings of insecurity. People may also be unwilling to give up their control over their environment, lacking in trust towards unfamiliar technology. Interactive architectural systems' ability of self-improvement may cater to reduction of iA's potential negative effects on human psychology and culture. However, this can happen only through the actual deployment and operation of iA systems and their co-evolution together with their human inhabitants, resulting in likely mutual adjustments.

Continuous adaptations promise reduction of involved costs along with increase of performance. However, dynamic systems require substantially more advanced technology, are more vulnerable to malfunctioning and require significantly more maintenance than static ones. Technological advancements lead to reduction of these costs, however, the overall balance of costs of traditional versus interactive architecture, at least on short term, might still be highly advantageous to traditional architecture. However, this discrepancy may be expected to be reduced and eventually inverted, once iA technologies become further developed and optimised.

It has been attributed to Henry Ford to having said that had he asked people what they wanted, they would have answered "faster horses", not cars, yet clearly after cars were put on the market they became a highly desired product, quickly replacing the use of horses as a mode of transportation. Similarly, there is not yet a direct demand for interactive architecture, as there cannot be much demand for something has not yet been fully developed and has not made its way to public awareness. However, there is a high demand for features, which iA is promising to offer, being, at least hypothetically, able to solve most of the pressing problems that traditional architecture is facing today.

6.4. Challenges and opportunities

Assuming that the promises of iA outweigh the risks, the greatest challenge is to develop a working interactive architectural system. This challenge has been put forward already in 1967 by Warren Brodey, who wrote "Can we now teach our machines or environments first complex, then self-organizing intelligence which we can ultimately refine into being evolutionary? The first answer to this question is that we do not know how. But if we do not know how to create a complex self-' organizing evolutionary environment, we can at least begin assembling the ingredients and concepts that can, by trial and error, produce better tools. If the task is impossible we shall at least learn why.“[1] Since then many attempts were made. Although none has yet fully succeeded in making a truly self-evolving building system, the revival of the trend in recent decade driven by increased power and decreased cost of digital technologies brings new hope. However, the challenge of creation of interactive architecture is manifold. Three main aspects of this challenge are methodology, technology and operation.

a) Methodology

Traditionally established approaches to design and creation of architecture aren't suited for developing dynamic architectural systems. It is uncommon for architecture to be designed and built with its own change in mind. Typically all active systems in buildings are installed as additions to otherwise static building structures, supplementing architectural design, rather than becoming its innate constituents. Therefore, creation of interactive architecture requires a new design methodology, where buildings are intentionally designed and built as dynamic systems. As Biloria observes: "Architectural design that emphasises "softspace" over "hardspace" is a little like "software" design rather than "hardware design" in computer terminology, where "hardware" refers to the physical machine and "software" refers to the programs that animate the machine."[2]

Along with new methods, new tools may need to be developed in order to facilitate dealing with the complexity of designed systems. Latest trends and researches in the domain of computer aided architectural design provide promising prospects.

b) Technology

Interactive architecture requires numerous technological advancements in order to be feasible. Although theoretically interactive architectural projects such as Cedric Price's "Fun Palac" were possible to be built with technology available in the 1970s, the high costs and a high number of technical constraints even today provides a substantial drawback. Easy to use and affordable technical solutions need to be found in order to facilitate further developments in iA.

Various technical developments, especially those originating from electronics and information technology, namely electronic sensors, microcontrollers and actuators, give prospect for interactive architecture to be realizable in its full scope. Electronic devices such as mobile phones, computers, entertainment systems, or even household appliances already surround us everywhere. Already in 1996 over 90 % of world's microprocessors were embedded not in computers but in common household appliances and products[3]. Yet, despite high proliferation of electronic devices in out environments, there is still not enough interconnectivity among them, and marginal relation between those devices and architectural spaces. Further development of such interconnectivity, dealt with by the domain of ubiquitous computing and its sister concept "the internet of things", may open up new paths for development of spatial interactivity. It may also gradually lead to cultural changes that would allow common acceptance of interactive built spaces, along with emergence of simple and intuitive interfaces for that interaction. The technology required for sensing, processing and actuating needs to be embedded in architectural materials. On that front, new developments in "smart

[1] Brodey, 'The Design of Intelligent Environments: Soft Architecture'.

[2] Nimish Biloria, *Adaptive Corporate Environments: Creating Real-time Interactive Spatial Systems for Corporate Offices Incorporating Computation Techniques* (Delft: TU Delft, Faculty of Architecture, 2007), 239.

[3] David Kline, 'The Embedded Internet', *Wired*, October 1996.

materials" lead to emerging availability of materials capable of inherent transformation. Computer numerically controlled (CNC) fabrication[1] allows to quickly produce customised replacement elements at costs comparable to mass production.

c) Operation

Interactivity of built spaces will require humans to culturally accept the lack of direct control over every aspect of changes in spaces we use. In return, it will provide humans with potentially infinite amount of less direct influence over a vast range of spatial qualities of spatial environments. Nevertheless, reaching full reliability of iA's operation and associated cultural acceptance of interactive architecture may require considerable amount of research, experimentation and gradual build-up of experience.

Despite aforementioned reservations, theoretically at least, reliability of interactive systems should ultimately become much higher than of static or automated ones. This is due to the fact that interactive systems have greater capability of adaptation to unforeseen conditions. A building with dynamically adaptive structure could be able to actively counterfeit changing external environmental forces, or even natural disasters. Numerous active solutions are already used in skyscrapers raised in areas with high probability of earthquakes and while employing latest technical achievements are being brought to their use, many more such solutions are in current development.

By influencing the development of iA systems, either explicitly or through daily activities, its inhabitants will become co-designers of architecture. The role of an architect is bound to change, into someone who sets iA systems in motion, rather than determining its final outcome. A different role for an architect can also be seen as a specific kind of user being a kind of a care taker of the system, whose feedback can be more specific than of an inhabitant.

In this way the design and operation of interactive architecture occur simultaneously. Similarly, the activity of building construction is to be a continuous process, where fabrication and installation of new components or removal of unnecessary ones can occur at any time throughout building's operation.

6.5. Problems

Since first cybernetic art installations of Schoffer and Pask that dealt with creation of interactive spatial environments, many artistic and architectural experiments have attempted to bring us closer to creation of interactive architecture. In disconnection with such creative experiments, in the domains of building automation and ubiquitous computing, technologies and standards have been developed in order to allow creation of systems of interconnected "smart" devices operating within built spaces.

Despite these developments, to date there are no established methods, models or standards that would be applicable to creation of interactive architecture. Much of the explorative work is performed in an ad-hoc manner, focusing on just selected aspects of spatial interaction rather than approaching the problem comprehensively and systematically. On the other hand, work in the field of building automation is highly specific in its application and does not address architectural problems, while also being too specific and constraining to serve as foundation for architectural experimentation.

Clearly, a generic iA system framework is missing that would allow for integration of various currently disjointed efforts in order to advance development of iA. A shared platform is needed to further systematically develop, test, validate and potentially deploy iA solutions that could operate reliably, safely and efficiently in real built context.

[1] Lisa Iwamoto, *Digital Fabrications: Architectural and Material Techniques*, 144 p. (Princeton Architectural Press, 2009).

The process of spatial creation is currently encapsulated in three phases of design, production and construction, and operation of buildings. Interactive architecture requires each of these phases to be performed simultaneously and throughout the entire lifecycle of a building, while giving the key role in this process to building inhabitants and stakeholders. A system framework for iA needs to inherently support such simultaneity.

Approaching building spaces as dynamic systems requires new design methods and design tools. New strategies have to be formulated and explored in order to lay ground for future developments.

III. Complex adaptive system view on iA

Summary:

The third chapter follows a postulate for an integrated approach to design, creation and operation of interactive architecture. This approach is based on considering interactive architecture to be a complex adaptive system constituted of autonomous agents, forming an actor-network of living and non-living entities. The chapter is concluded by a research strategy towards attaining such approach.

The systemic understanding of traditional architecture and of interactive architecture is firstly thoroughly scrutinised and discussed in detail on the ontological level. Following that understanding, in synchrony with examination of common architectural praxis, it is concluded that current design methods, building procedures and culture of managing buildings constrain creation of interactive architecture. These observations consequently lead to asserting the need for new strategies, methods, instruments, techniques and open building operation scenarios to be developed in conjunction with incremental specification of an integrated design[1] framework for interactive architecture. In answer to this need, a research methodology is chosen to permit constructive development of such design framework.

The chapter begins with an overview of concepts that stem from the general consideration for architecture to be a complex system made up of interrelated material objects, people and other living organisms, and immaterial (non-embodied) entities (section 1.). Successively, the concept of architectural agency (capacity of architecture to act in its environment) is analysed in context of multi-layered architectural complexity and consequent augmentation of the agency of architecture and resultant local and global adaptations (section 2.). Problems resulting from presented worldview, which are faced by architectural designers, are discussed. Consequently, applicability of design principles, methods and tools that are traditionally employed in architecture and other design domains to design of complex architectural adaptive systems is questioned and possible alternatives are discussed alongside latest trends in architectural design tools (section 3.). The findings of the chapter are summarised, discussed and reflected back on the research framework (section 4.).

[1] Kiel Moe, *Integrated Design in Contemporary Architecture*, 1st ed. (Princeton Architectural Press, 2008).

1. Architecture is a complex system

The previous chapter has scrutinised the general relationship between people and architecture and the multitude of its aspects. This chapter breaks down that relationship between two largely abstract notions into specific relationships between individual human beings and individual constituents of cities, buildings and other architectural artefacts.

We may commonly think of any building as being a singular, static entity. Yet we also know that such way of looking at a building is a simplification of reality, where in its strictly material sense, every building is a system of interconnected, discrete physical components forming the spatial structure; a system consisting of many interrelated material elements forming what we perceive as a larger whole.

Inhabitants of a building themselves, performing activities within building spaces, are, equally to its material components, inherent to any building system, that is, they operate within that system's boundary. Eventually, every architectural system is open, being strongly influenced by factors from outside of its boundary, such as by climate or by neighbouring built environments. Such understanding of architecture (as a system of interrelated parts) can be further extended by consideration of a commonly overlooked type of building system's material elements; "spaces". Building spaces are air-filled voids, which other building components frame and organise, and which are inhabited by building's occupants. Some elements of architectural systems are in fact collections of multiple components. Others are abstract entities established by convention, e.g. ownership.

Understanding of architecture as being a complex system built up of such diverse elements requires an appropriate view on architecture, which has gradually developed within modern architectural theory over past decades.

1.1. Systems, complexity and architecture

A system is a set of elements and binding them functional and structural relationships, as a whole exhibiting some kind of behaviour[1]. The contemporary understanding of the concept of a system and the wide domain of "systems thinking" can be traced back to the "General Systems Theory" of Ludwig von Bertallanfy[2]. Systems can be defined as closed or open, depending whether matter and energy can cross their boundaries and whether elements can be added or removed from the set. Bertalanffy's position is that closed systems with impermeable boundaries don't exist in reality; every observable system is open.

The foundations for the contemporary understanding of the performance of systems have been laid by research in the domain of cybernetics, the study of regulatory systems. Norbert Weiner is the pioneer of this discipline, who has provided grounds for mathematical models of systems[3] and defined the notion of feedback and feedback loops in systems, which are continuous action-response cycles, where positive feedback increases the magnitude of its cause and negative feedback attenuates its cause and effect.

Since its outset, the systems science has been employed to understand phenomena of high complexity, which means dealing with systems consisting of a very high number of interrelated parts. Warren Weaver, has been one of the first to identify theinterdisciplinary

[1] Alexander Backlund, 'The Definition of System', *Kybernetes* 29, no. 4 (June 1, 2000): 444–451.

[2] Ludwig von Bertalanffy, *General System Theory: Foundations, Development, Applications*, Revised (George Braziller, 1976).

[3] Wiener, *Cybernetics, Second Edition*.

domain of complexity science[1], and through his work for the Rockefeller Foundation he has promoted many scientists engaged in their work with complex phenomena. Some of the most notable work in developing end employing rigorous scientific methods for dealing with complex systems that followed between the 1960s and 1980s includes Ilia Prigogine's identification of the mechanisms of self-organisation[2], Benoine Mandelbrot's and Edward Lorentz' work on chaotic systems, Friedrich Hayek's analysis of complex mechanisms in economy in relation to politics and society and foundations of complex systems modelling, Niklas Luhmann's applications of systems thinking to sociology, Stuart Kauffman's work on understanding complex biological systems and John Holland's understanding of mechanisms of adaptation and evolution in complex systems, leading to the definition of complex adaptive systems[3] and providing foundations for the domain of evolutionary computing. Since its early days, complexity science has become widely recognised and applied in many disciplines[4]. Developments in computer science have provided unprecedented instruments for simulation and creation of artificial complex systems. For detailed reference, the contemporary outlook on complex system theories and involved mathematical models is elegantly summarised in the "Foundations of Complex-Systems Theories" of S.Y.Auyang[5].

Developments in systems and complexity science have had an influence on architectural design. Clearly, examples discussed in sections 3.2 and 3.3 of chapter II are tightly intertwined with developments in systems thinking and cybernetics. In cases such as projects and written work of Cedric Price, Gordon Pask, Nicholas Negroponte, Archigram, Yona Friedman, Charles Eastman or the Metabolists, treating architecture as a system of elements leads to considerations of various new possibilities in enhancing architecture's adaptability by increasing its ability to transform itself or to be transformed though external action. Yet, those ambitious developments have remained marginal to the broad architectural discourse. In wider architectural practice, in 1960s and 1970s the notion of a "building system" became synonymous to a system of modular, prefabricated and repetitive building components, in effect associating the notion of system in architecture with architectural homogeneity and consequently with low quality building. The understanding of architecture operating as a system, has been limited to analysis of human performance within predefined function, rarely considering for the larger context, or more complex factors. Bachman provides a generic, yet arguably incomplete, overview of "encounters of architecture with complexity"[6], which can be used as a general reference. The following paragraphs, are thus not an attempt to provide another overview or classification, but to trace a number of problems involving both architecture and complex system theories, while also relating to other domains where considered relevant.

[1] W. Weaver, *A Quarter Century in the Natural Sciences*, The Rockefeller Foundation Annual Report (The Rockefeller Foundation, 1958).

[2] Ilya Prigogine and G. Nicolis, *Self-Organization in Nonequilibrium Systems: From Dissipative Structures to Order Through Fluctuations*, First Edition (John Wiley & Sons, 1977).

[3] John Holland, *Hidden Order: How Adaptation Builds Complexity*, First Edition (Basic Books, 1996).

[4] M. Mitchell Waldrop, *Complexity: The Emerging Science at the Edge of Order and Chaos*, 1st ed. (Simon & Schuster, 1992).

[5] Sunny Y. Auyang, *Foundations of Complex-system Theories: In Economics, Evolutionary Biology, and Statistical Physics* (Cambridge University Press, 1998).

[6] Leonard R. Bachman, 'Architecture and the Four Encounters with Complexity', *Architectural Engineering and Design Management* 4, no. 1 (March 2008): 15–30.

Systems-thinking has had a stronger resonance in architectural debate in relation to problems of the urban scale[1]. The early seminal model for dealing with urban systems has been presented in 1960 by Kevin Lynch in "The Image of the City"[2]. This model abstracts any city system to be an assembly made out of five types of elements, namely: paths, edges, districts, nodes and landmarks. Clearly, despite its usefulness for general description of city structures, Lynch's model is very reductive. Although it accounts for continuous change in urban systems, it does not represent processes occurring in a city, nor does it reflect city's' true diversity, heterogeneity, history of development and the possibility of numerous interpretations.

A more in-depth viewpoint towards the understanding of urban systems and their intricate processes is described in Jane Jacobs' "Death and life of great American cities", where the focus is not on a rigidly classified system of components of a city, but rather on understanding of very complex, heterogeneous processes taking place in urban systems. In her text Jane Jacobs brings up Warren Weaver's definition of "organized complexity" ("problems which involve dealing simultaneously with a sizeable number of factors which are interrelated into an organic whole"[3], situating organised complexity between order and disorganized-, or chaotic complexity) and uses that notion in her argument for the new way of understanding cities and their spaces[4]. She provides a city park as an example of a public space that is in a state of continuous development. A park is in fact a collection of many parts, which all, in different ways, continuously affect park's users and at the same time the park's performance as a whole is being affected by ways in which park visitors act and by ways in which the wider context of the park is changing, making the park continuously altering in the way it functions; "no matter what you try do to it, a city park behaves like a problem in organized complexity". In all kinds of public spaces, in certain conditions, a simplest element of urban furniture may become the strong catalyst for many complex social activities, whereas, in just slightly different external circumstances, even the most sophisticated urban facilities may strive to attract life to the premises of the very same public space. Generalizing this phenomenon to larger scales of architecture, Jacobs writes; "Objects in cities – whether they are buildings, streets, parks, districts, landmarks, or anything else – can have radically differing effects, depending upon the circumstances and contexts in which they exist". To provide a guideline for understanding such complex spatial systems, she proposes three main rules "1. To think about processes 2. To work inductively, reasoning from particulars to the general (...) 3. To seek for "unaverage" clues, involving very small quantities, which reveal the way larger and more "average" quantities are operating". This marks an on-going paradigm shift[5] in architectural and urban design praxis. Rather than thinking in a top-down manner and understanding spatial environments as permanent and definitive (which has dominated western architectural design for centuries), Jacobs urges to do the opposite, to see architecture as an on-going process and architectural design as induction to such process, performed in a bottom-up manner through (possibly simple) interventions into a complex system which is in perpetual motion. This strategy means finding the most primary, basic elements that constitute the spatial environment and gradually understanding how greater phenomena emerge from their aggregation.

[1] Michael Batty, *Cities and Complexity: Understanding Cities with Cellular Automata, Agent-Based Models, and Fractals* (The MIT Press, 2005).

[2] Kevin Lynch, *The Image of the City* (The MIT Press, 1960).

[3] Weaver, *A Quarter Century in the Natural Sciences*.

[4] Jane Jacobs, *The Death and Life of Great American Cities* (Modern Library, 1993).

[5] Thomas S. Kuhn, *The Structure of Scientific Revolutions*, 1st ed. (University Of Chicago Press, 1996).

An approach which in many ways resonates with Jacobs' understanding of urban systems is Christopher Alexander's consideration for seeing architecture as a set of overlapping systems, a "semi-lattice", which was for the first time described in his essay "The city is not a tree"[1]. Alexander contrasts the tree and semi-lattice in that they are both "ways of thinking about how a large collection of many small systems forms to make up a large complex system", while a tree is entirely hierarchical, a sub-system has to be entirely contained in the larger system, while semi-lattice means that sub systems can go beyond the boundaries of systems they are contained in. Alexander claims that tree organisation common to modern urban systems results in the "lack of structural complexity (...) crippling our conception of the city", stating that "a living city is and needs to be a semi-lattice".

In line with this thinking, in place of defining a rigid taxonomy for an urban system Alexander proposes an openly defined set of patterns: small, overlapping sub-systems operating within and sometimes beyond the larger system of the city. Presented by him "pattern language"[2] consists of 253 elaborately described patterns of complex relations, dependencies and interactions between people and space. These patterns operate on various scales and can be tightly interwoven as well as nested in each other. Despite the vast and diverse account of described patterns and the comprehensiveness of their discussed aspects and features, Alexander himself treats his exhaustive classification not only as non-definitive, but as entirely subjective and open to any further alteration. With this, he urges everyone to make their own modifications to his "pattern language", or even to formulate entirely new "pattern languages", hoping for the future when "(...) every society which is alive and whole, will have its own unique and distinct pattern language; and further, every individual in such a society will have a unique language". Alexander's understanding of architectural and urban systems in relation to design process theory is explained in "Notes on the synthesis of form"[3] and "The timeless way of building"[4], where the design as means of architectural system adaptation is among others thoroughly discussed. Despite their potential, Alexander's theories and proposed methods have not profoundly affected architectural praxis. Arguably the approach has not been broad enough, or the identification of the limited number of existing and successful patterns has not been enough on itself to allow creation of thoroughly successful architectural interventions, although, "The Oregon Experiment"[5] can be seen as a partly successful attempt in application of Alexander's model and methods in practice and creating a spatial intervention, which is highly participatory in its mode of development.

Other post-modern architectural theorists and architects have also attempted to capture the phenomenon of architecture as being part of a complex system (in those cases, mostly understood through the prism of the cultural and historical context) and its expression in resulting architectural forms. Among those, in "Complexity and Contradiction" Robert Venturi postulated architecture that "must embody the difficult unity of inclusion rather than the easy unity of exclusion". He urged that in "the growing complexities (and contradictions) of our functional problems (...) even the house, simple in scope, is complex in purpose if the ambiguities of contemporary experience are expressed"[6]. However, in Venturi's own projects this elaborate theory has been reflected merely in the formal language. Post-modernist expression of architectural forms, in general, became limited to simplified transformations and re-compositions of historical details. It also became detached from the structure of

[1] Christopher Alexander, 'City Is Not a Tree', *Design*, 1966.

[2] Christopher Alexander, *A Pattern Language: Towns, Buildings, Construction* (Oxford University Press, USA, 1977).

[3] Christopher Alexander, *Notes on the Synthesis of Form* (Harvard University Press, 1964).

[4] Christopher Alexander, *The Timeless Way of Building*, Later printing (Oxford University Press, 1979).

[5] Christopher Alexander, *The Oregon Experiment*, 1ST ed. (Oxford University Press, USA, 1975).

[6] Robert Venturi, *Robert Venturi: Complexity and Contradiction in Architecture*, 2nd ed. (The Museum of Modern Art, New York, 1977).

buildings, either creating "decorated sheds" - additional layers to otherwise mundane structures, or "ducks", where buildings become symbols in their entity; both these types glorified by Venturi and Scott Brown in "Learning from Las Vegas"[1]. Yet, beyond the formal superficiality, the valuable lesson from architectural postmodernism over modernism is that "the new" need no longer entirely negate and replace the old, but it can build up upon it and transform it instead.

Clearly, it required time for the paradigm shift in the understanding of architecture as a complex system to mature. Charles Jencks refers to this more mature movement as "new modernism"[2], which adds another layer to the understanding of modern architecture in place of fuelling itself on plain critique of modernism. Peter Eisenman's work is among the best early examples of that trend.

Rem Koolhas is named as another representative of the "new modernism". In Rem Koolhas' "Delirious New York", Koolhaas delves on the interrelationship between the metropolitan culture of New York and its architecture[3]. Rather than taking a designer's standpoint, he observes and analyses the unavoidable mutual dependency between people and artificial city fabric, and the vast amount of relations that have shaped New York City as a complex organism. This publication marks the transformation of the superficial postmodern praxis in architecture into a more mature understanding of spatial environments. In Reiser and Umemoto's words, contemporarily we understand that: "architecture falls into an intermediate category between matter and events. It is a modulator."[4] Of many contemporary noteworthy texts that relate to this understanding, the works of Henri Lefebvre, among others in "The production of space"[5], deal with the understanding of architectural space being a social construct which resonates in later works of Manuel Castells, expressed further in Bill Mitchell's "City of Bits"[6]. In clear distinction from Castells, Bruno Latour, extends these networks to include objects (with consequences more thoroughly discussed in later sections), whereas, in the writings of Manuel De Landa, architecture indirectly comes across as a spatial manifestation of continuous, non-linear historical processes[7], seen through the prism of the ontology of Gilles Deleuze, and in Deleuze's terms, architecture becomes the rhizome[8] of interweaving matter, life and culture.

The understanding of architecture as an emergent phenomenon; a manifestation of a complex system operating across historically established domains (alternatively considered, following Bruno Latour, to be a subset or a reduced view on the traceable and boundary-less actor-network of humans and things) is thus an established direction, which many voices in contemporary architectural theory and praxis are already following. In this way, architecture can be seen as a complex process which can potentially involve any kinds of actors in an infinitely traceable network, for understanding and working with which architects have limited means[9]. In this view, a (complex) system is a reduction of such network into a manageable and constrained whole. More questions begin to rise, however, since understanding of

[1] Robert Venturi, Steven Izenour, and Denise Scott Brown, *Learning from Las Vegas - Revised Edition: The Forgotten Symbolism of Architectural Form*, Revised (The MIT Press, 1977).

[2] Charles Jencks and Karl Kropf, *Theories and Manifestoes of Contemporary Architecture*, 2nd ed. (Academy Press, 2006).

[3] Rem Koolhaas, *Delirious New York: A Retroactive Manifesto for Manhattan* (The Monacelli Press, 1997).

[4] Jesse Reiser, *Atlas of Novel Tectonics*, 1st ed. (Princeton Architectural Press, 2006).

[5] Lefebvre, *The Production of Space*.

[6] William J. Mitchell, *City of Bits: Space, Place, and the Infobahn* (The MIT Press, 1996).

[7] Manuel De Landa, *A Thousand Years of Nonlinear History* (Zone Books, 2000).

[8] Gilles Deleuze, *Thousand Plateaus* (Continuum International Publishing Group, 2004).

[9] Bruno Latour and Albena Yaneva, 'Give Me a Gun and I Will Make All Buildings Move: An ANT's View of Architecture', in *Explorations in Architecture: Teaching, Design, Research* (Basel: Birkhäuser,, 2008), 80–89.

spatial complexity in itself does not provide all the answers as of how to create architecture appropriately operating within that complexity. Yet, before the creation of architecture as a complex system can be thoroughly discussed, the systemic nature of architecture needs to be analysed in greater detail, starting with the discussion on the taxonomies of architectural system components and their implications.

1.2. Architectural system components

Defining the singularities of components forming architectural systems is a question of both convention and purpose and inherently involves a reduction of what that definition describes. From the generic standpoint, an architectural system component can be any kind of entity; material or immaterial, human or non-human, observable on macro- or micro-scale.

In order to discuss the specificity of understanding of architecture as a system, to define the vocabulary of related phenomena and to discuss consequent problems and concerns, an attempt to identify most common components of architectural systems has to be made. It is not claimed that following attempt to discern such taxonomy of architectural components is the only "true" categorisation. On the contrary, it is to be seen as an open proposal, presented with the intention of being further freely extended, reduced or otherwise altered.

a) Material architectural components

Clearly, the most common way to look at architecture is to see it as an assembly of material entities. The most general way to classify material components of architecture is by their scale. The largest component of what is typically considered to be of architectural "kind" is a city, or any other form of human settlement, such as "village", "town" or "encampment". Yet, even such large entity as a city, can be a part of even a larger entity, such as a "region" or a "country", although those are not commonly considered as belonging to the domain of architecture. Moving down in scale, an "urban block", "park", "city square", "building", "room", "wall", "furniture piece", "brick" are some other architectural components we may encounter on the way. Such "zooming in" on the scale of architectural components may theoretically be continued (as in the film "The Powers of Ten" by Charles and Ray Eames, arguably a misleading approach), breaking down granularity of materials, particles, atoms and beyond into the quantum scale. Any of such nested components can be considered to be part of an architectural system, although going beyond the scale of material entities perceivable by humans may not be considered to be "architectural". In consequence it can be observed that the commonplace definition of what is considered to be a material architectural entity has to be perceivable by a human being. Both a brick, when looked at up close, and a city, when looked at from a distant perspective, are observable as entities without a need of any instruments or representations. An atomic particle and a region go beyond human ability to be directly perceived and require an instrument such as a microscope or a map in order to be empirically observed and described. What's more, even a city requires an action of stepping beyond its boundary to be seen as an entity. Similarly, observing a park from within, means seeing individual trees, benches and joggers, rather than the park as whole. Being inside a building results in noticing individual walls, appliances, furniture and inhabitants, rather than seeing the building as one entity.

From such perspective, a standpoint can be taken to use the scale of direct human perception as leading in the framing of material components in an architectural system. From this standpoint a building can be a relevant architectural system component, as it observable as a distinguishable entity by a person walking in the street. A wall in that building is a distinguishable entity for a person who enters that building. A brick is an entity clearly distinguishable for a mason raising the wall, but may lose its identifiability for an inhabitant of the building, who only notices a plastered wall.

Following human perception, architectural components can be classified based on their geometry and be subdivided into volumes, surfaces, linear elements and nodes. Volumetric elements are characterised by all their dimensions being equally large, forming blocks of impenetrable material. Surfaces have their two dimensions significantly larger than the third one. Linear elements' one dimension is significantly greater than the other two, while nodes have all dimensions reasonably small comparing to other surrounding elements. Geometrically, any architectural geometry can be reduced to a combination of these basic geometric elements. Noteworthy, as pointed out by Gestalt psychology, assemblies of entities of one kind, such as points, can be perceived as another kind, such as plane or volume.

Geometric distinction of building components often corresponds to their functional role. The fundamental role of building components is organisation of space. Typically wall, roof and floor surfaces frame architectural spaces, however this convention can be creatively challenged. The function of these surfaces is protection from precipitation, thermal insulation, protection from sun, and potential dangers of the outside world, as well as acoustic and visual separation of distinct architectural spaces. Another function of architectural components to provide structural for other components and each other, resisting tensile, compressive and bending forces occurring in the structure. Architectural matter can also be fitted with installation networks and devices providing highly specific and distributed functionality.

On the urban design scale, individual building components lose their relevance. It is the entire buildings and their properties being the convergence of all building parts that become primary components of an urban plan, together with other elements of urban infrastructure.

Accordingly, it can be proposed to treat entities that are assemblages of other entities are as equally valid components of the system, possessing a certain relation to their nested-sub-components, but being definable independently. In this manner "a city", "a building" and "a brick" can be treated as interrelated material components of the same system.

b) "Space"

In architectural context, the notion of *space* appears as an ambiguous term. Physically, it refers to the property of our universe organising matter and energy in terms of their proximity. Architecturally, "space" is usually referred to as the void, typically filled with air (but possibly with other gas or liquid), framed by solid material. The important quality of architectural "space" is that it can be occupied by humans. It is in this context, rather than the physical one, "space" can be considered as a constituting component of any architectural system. On the other hand, space as a physical phenomenon (to be from now on used without quotation marks to point out this meaning), in conjunction with time, can be seen as an organising structure of architectural systems. It is in space and time that architectural systems unfold and transform.

Architectural "space" can be considered to be the primary "substance" of architecture. In words of the first Pritzker Prize winner Philip Johnson "all great architecture is the design of space that contains, cuddles, exalts, or stimulates the persons in that space"[1] (who earlier humorously stated that "architecture is the art of how to waste space"[2]). Synthesising this and other countless definitions, it can be broadly stated that architecture is the organisation of "spaces" by humans. Clearly, architects do not predominantly occupy themselves with studying the physical nature of space itself though (however, it is a significant part of the theoretical debate on architecture), but rather usually settle for perceiving "space" as a uniform void filled with air whose qualities are defined by its boundaries. To a non-specialist reader, Encyclopaedia Britannica describes this phenomenon as following; "Space, that

[1] Philip C. Johnson, *Writings*, 1st Edition. (Oxford University Press, 1979).

[2] New York Times, Dec. 27, 1964

immaterial essence that the painter suggests and the sculptor fills, the architect envelops, creating a wholly human and finite environment within the infinite environment of nature. The concept that space can have a quality other than emptiness is difficult to grasp. When a building is entered, floor, supports, walls, and a ceiling are seen, all of which can be studied and perhaps enjoyed, while the space, in the sense that one is accustomed to think of it, is void: the absence of mass, filled by air. But spatial experiences that express something are common to everyone, though they are not always consciously grasped."[1] Architects typically define distinct "spaces" by setting their boundaries, thus also separating them from other "spaces". As Sigfried Giedion has put it when describing Egyptian pyramids and pantheon as some of the first prominent examples of architecture "It is the interaction between volumes which gives full orchestration to the first architectural space conception"[2]. These boundary volumes may be impenetrable by human perception, (such as massive stone walls in ancient architecture), may allow some penetration (e.g. glass allowing light to pass through) may be a boundary of no physical manifestation but appearing to humans through indirect mechanisms of perception (Gestalt Principles of Perception lay foundations for understanding basic rules of spatial perception) or through specific sensations (e.g. smell, or temperature), may be only detectable by artificial devices (e.g. area in range of the computer wireless network, magnetic field or irradiated zone) or even be not empirically distinguishable at all, but set through a convention or agreement (e.g. by defining its geographical coordinates).

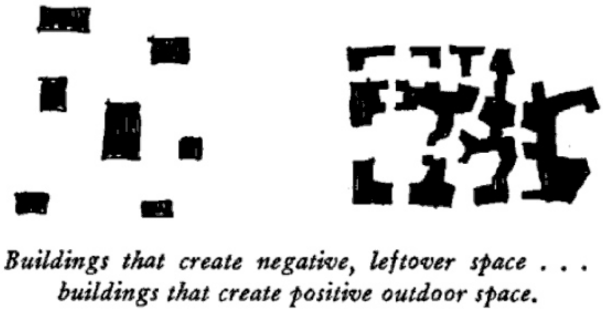
Typically architectural "spaces" are described by means of three dimensional Euclidean geometry, with orthogonal grid being the simplest organisational principle to apply in that geometric paradigm. Although architectural geometry is not the focal point of this work, certain aspects of presented research may demonstrate the potential for non-Euclidean geometric descriptions of architectural "spaces". (Nevertheless, all explorations will ultimately remain re-mapped into three dimensional Euclidean models, for the clarity of demonstration.)

In architecture, a fuzzy distinction can be made between positive and negative "spaces", where positive spaces are perceived as voids carved out of solids and negative "spaces" are perceived as open voids without clear boundaries. Inverted depictions of "space" allow us to more explicitly visualize the spatial voids that architecture creates. Those voids (physically filled with air) together with the elements that define them, create places, subjectively identifiable locations in "space". "Spaces" may have functions assigned to them; ways in which spaces are culturally intended to be used. Traditionally, functions in buildings are considered as fixed and pre-defined. Contemporarily, such notion of a function is questioned[3] and changing or flexible, adaptable functionality in buildings is commonly introduced.

[1] 'Architecture :: Space and Mass -- Britannica Online Encyclopedia', accessed March 17, 2010, http://www.britannica.com/EBchecked/topic/32876/architecture/31847/Space-and-mass.

[2] Siegfried Giedion, *Space, Time and Architecture: The Growth of a New Tradition, Fifth Revised and Enlarged Edition*, 5th ed. (Harvard University Press, 1967).

[3] Jonathan Hughes and Simon Sadler, *Non-Plan: Essays on Freedom, Participation and Change in Modern Architecture and Urbanism* (Architectural Press, 2000).


Buildings that create negative, leftover space . . .
buildings that create positive outdoor space.

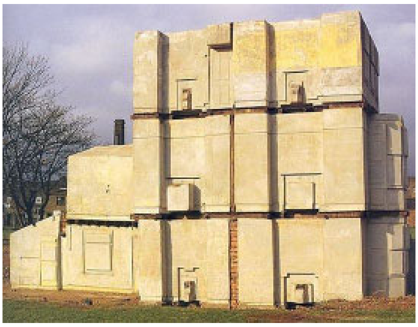


Fig.13. Positive and negative "space" diagrams of two typical city space patterns from Christopher Alexander's Pattern language[1]. Alexander argues that "space" is positive when it forms distinct clusters and negative when it flows without explicit boundaries "Outdoor space is negative when it is shapeless, the residue left behind when buildings – which are generally viewed as positive – are placed on the land. An outdoor space is positive when it has a distinct and definite shape, as definite as the shape of a room".

Fig.14. House, sculpture by Rachel Whiteread, which is a cast of the inner "space" of a later demolished Victorian house, showing a solid model of a positive space, 1993

c) Humans

The combination of "spaces" and framing them material artefacts delivers affordances for particular activities to be performed within architecture. Traditionally, human inhabitants performing such activities in architectural systems are considered to be external to those systems. Thus, functions in architectural systems tend to be defined a priori to the actual deployment and operation of architecture[2].

However, humans performing inhabiting architecture are physically surrounded by the "space" of architecture, thus remain within the physical boundary of an architectural system. Therefore, it is convenient and logical to consider human inhabitants as components architectural systems, despite the fact that architectural and social systems are commonly considered to be separated (with the exception of the Actor-Network social theory). Treating inhabitants as active parts of the system, living in a continuous interplay with other system components, including material and space, as well as interacting other users, changes that perspective. Inhabitants become actors involved in continuous re-definition of architecture they inhabit, becoming inherent parts of it.

From this perspective, the distinction between an inhabitant, builder, stakeholder or designer blurs. An inhabitant can be equally a creator, controller or designer of architecture he or she inhabits, or has an influence on. Inclusion of humans as components of an architectural system also allows for much thorough study and engagement in such system's adaptation, occurring always as interplay of human and non-human components.

d) Living matter

The role of non-human living organisms in architecture cannot be underestimated. Microorganisms, plants and animals can have strong influence and at times critical influence on the functioning of architecture.

[1] Alexander, *A Pattern Language*.

[2] Vic Callaghan, Graham Clarke, and Jeannette Chin, 'Some Socio-technical Aspects of Intelligent Buildings and Pervasive Computing Research', *Intelligent Buildings International* 1 (2009): 56–74.

In certain scenarios an animal can be treated as an inhabitant of architecture with similar status to humans. Inhabitation of space by a house pet, a bird in a birdhouse or a zoo animal, may directly dictate the architectural qualities to be intentionally created by humans for these animals. However, an even larger group of animals inhabit and transform architecture without an explicit intention of humans. Rodents or beetles quite directly transform buildings to their own needs, often leading to conflicts with humans treating them as pests.

The role of vegetation in architecture can be equally ambiguous. Trees and other plants are often planted in synchrony with architecture, being treated as part of its materiality and providing many architectural qualities from shading to decoration. On the other hand, in many cases, plants growing against the will of human inhabitants are treated as pests.

Since microorganisms are not directly observable by humans, phenomena they cause are often directly attributed to entities that are being transformed by them, such as e.g. decay of wood. Nevertheless, microorganisms are the most ubiquitous form of life on Earth and if it was not for their scale, they could be taken into account as components in an architectural system of their own right. Similarly to other living organisms their role in architectural systems may be seen as destructive (mould, fungi, material decay), nevertheless in larger perspective microorganisms are the foundation of any ecosystem and without them life on earth, or the very functioning of the human body would not be possible.

e) Non-embodied components

In many cases architectural systems may contain clearly identifiable components that are not embodied in any particular material entity. They are abstract concepts that are integral to the system and relate and effectuate its many components. Typically, non-embodied components describe properties or constraints on other components. So can be the “climate”, which describes a property of a certain region. A climate is less than an aggregate of all constituting it components, but describes an aspect of such aggregate. Similarly, laws, as well as any other cultural conventions (including semantic entities, signs, values, meaning, memes) can be seen as components in their own rights that emerge out of actions of humans in the society, but are not embodied in any explicit material component or aggregate of these components. These components emerge out of an assemblage of a set of components, but are independent of these assemblages. Generally, the non-embodied components of a system can be seen as abstraction of qualities, forces or properties of a set of system components that are shared by these components.

1.3. Relations in architectural systems

Every system consists of components and relations between these components, in which definition of both; components and relations, always involves reduction of reality that the given system describes. The dictionary definition of “relation” states that it is “an aspect or quality (as resemblance) that connects two or more things or parts as being or belonging or working together or as being of the same kind”[1]. Consequently, a relation designates how two or more objects are connected to each other in respect to their mutually referencing descriptions or in respect to their mutual influence and invoked by this influence transformation. Within such broad definition, an attempt can be made to identify main contexts in which relations occur in consideration to the generality of architectural systems and based on that, to further discern and discuss several types of relations in such systems. The following classification is organised by outlining three fundamental types of relations, ordered by their

[1] ‘Relation - Definition and More from the Free Merriam-Webster Dictionary’, accessed June 27, 2011, http://www.merriam-webster.com/dictionary/relation?show=0&t=1309181372.

increasing complexity and accuracy of portrayal of real world phenomena. The first type of “descriptive” relations provides only the relative description among entities by referencing in that description one entity to the other, and is specified for one instance of time. In the second type “performative” relations describe of processes in which relative descriptions of entities change over time. The third type of “transformative” relations includes descriptions of how entities cause each other’s’ inner transformations. Within each type of relations specified in this general classification, relations are discussed by initially focusing on relations exclusively among embodied components, starting from relations between components of similar kind and consequently reaching relations between components existing on different scales of aggregation, in order to eventually include relations with and among non-embodied components.

a) Descriptive

Descriptive relations define the conditions and qualities of an entity in relation to another entity or a set of entities.

All physical characteristics of an entity can be described using descriptive relations to other entities. In daily human life many physical descriptions, such as geographical position, altitude, and orientation are done in reference to the surface of the Earth. Even weight is the effect that gravity of the Earth has on an object, which is linearly proportional to this object’s mass. Yet, the reference to Earth is only an intermediary used to formulate relative descriptions between individual entities. Rather than reference to the Earth as a whole, it is the specific geographical features on earth’s surface as well as entities populating this surface among which distances and orientations are in practice needed.

Traditionally, the role of architecture is seen as maintenance of physical relations between material architectural components in an equilibrium state and in reference to the surface of the Earth (or in rare examples such as cruise ships, in relation to a moving, but shared by all architectural components frame of reference). On the other hand, architectural adaptation inherently requires physical relations between material architectural components to be continuously altered.

Aforementioned properties of entities are all extrinsic, that is, they are defined as relationships to other, external entities. The other group of properties are those defined as intrinsic, describing qualities inherent to entities and not directly dependent on relations to external entities. Where properties such as position, orientation or weight are extrinsic, density, mass, temperature, are some examples of intrinsic properties.

It can be claimed that intrinsic properties of an object, where object is in fact also always a system of some sub-components, are a specific kind of relations that define that object[1]. If they point to other objects these properties are external relations. If they point towards sub-objects of the described object they are internal relations. Thus, also intrinsic properties can be described through relations, however these relations, unlike in extrinsic properties, don’t point outwards to external entities, but inwards, to entities that are sub-parts of described aggregate. In this way density would be a relationship between the material entity seen as a whole and its countless individual particles whose relative positions define the global density of the entire assemblage. Similarly temperature of a material entity, can be defined as a set of relations between that entity to its material sub-parts and their individual temperatures, where the global temperature could be defined as the weighted average of all the sub-temperatures. This example illustrates distinction between intensive and extensive types or intrinsic properties. Intensive properties are independent of object’s size or mass, or in

[1] Chris Swoyer, ‘Properties’, in *The Stanford Encyclopedia of Philosophy*, ed. Edward N. Zalta, Winter 2009, 2009, http://plato.stanford.edu/archives/win2009/entries/properties/.

other words they are averages of properties of all object's sub-objects. On the other hand, extensive properties are proportional to the system size and/or the amount of material in it; they are established by summation of corresponding extensive properties of its sub-systems.

As explained earlier, any object can be considered to be a system in its own right and other systems can exist that include some of that object's components, as well as other elements that do not belong to the given object, in effect forming a semi-lattice of sets. Consequently semantic relations can be brought into the picture in order to define how elements operating on different scale conceptually and semantically relate to one another. Relations of meronymy (being part of) and helonymy (having something as a part of itself) can form a very complex network operating across multiple nested and overlapping (semi-lattice) systems.

In the context of the relations of meronymy and helonymy, a property or attribute of an object can be seen as a specific kind of a relation. It can be intrinsic if it describes properties constituted within the object's systemic boundary (e.g. mass). In this case it is a relation between an object seen as a system and its sub-components. It can be extrinsic when it is defined by a relationship between the described object and other objects, outside of that object's boundary (e.g. weight).

A non-embodied component has been considered to be a component defining an immaterial quality in a system that is a property emergent out of an interaction between many entities, such as e.g. density, or any other immaterial entity that may have a representation, but is principally a mental human construct. In respect to non-embodied components, all relations among such components, and between embodied and non-embodied components can be referred to as virtual. Many semantic relations constituting a semantic network belong to this category.

A thin line exists between what is a virtual component and what is a component that is an assemblage of other material components. Both can be considered as existing only through interpretation of reality through human perception; however a physical assemblage, such as "a wall" always needs to contain some other material entities as its sub-parts, whereas a purely virtual components, such as "a building regulation" only contains virtual sub-components (i.e. a physical book describing a building regulation is not part of that building regulation, although it is related to it). On the other hand virtual components can be parts of larger physical components. A building regulation can be part of the legal system, which is a part of a social system, where a social system is in fact a system of material entities – humans and their habitats where the legal regulations are virtual components emerging from relations of humans and their habitats over time.

b) Performative

Descriptive relations provide a picture of the state of the system in a given instance of time. Even if some properties of system components (such as e.g. position relative to other components) change over time, they can often be described through unchanging relative descriptions (such as e.g. relative velocity and acceleration or sine wave oscillation), which can be done for most cases in closed systems (which are always reduced and never exist in reality). However, relations between elements in complex and open systems that are open to many and/or unpredictable inputs can change in ways that cannot be described with static relations. In those cases more complex "performative" relations need to be formulated in order to describe the given system not only in an instance of time, but also its capacity to perform and change over time.

Relations that can be called "performative" have to account for a force existing between entities. Similarly to description of spatial relation, force relation has a magnitude and a direction. Although the mechanical forces can be considered as symmetrical where in Newtonian mechanics action always equals reaction, the direction can be generally related to temporal causality and in some cases may be the question of assumed frame of reference. Physical forces, on particle scale referred to as “fundamental interactions” can be gravitational, electromagnetic (including visible light, infra-red, ultra violet, x-rays, gamma rays, microwaves and radio waves) as well as weak and strong nuclear. On human scale fundamental forces translate to most of perceivable properties of matter, which emerge only on larger scales. Because of this, it is practical to deal with non-fundamental forces on human scale, such as the normal force (direct force of touching material entities, perpendicular to the touching surface), friction, tension, elastic force, as well as pressure, stress and drag in continuous materials.

Directionality of performative relations is always dependent on a given frame of reference in time and space. When the frame of reference is removed, the direction loses its relevance. Transformations induced by relation are always opposite to each other and maintain equilibrium of matter and energy. The direction of a relation indicates the causality of transformation induced by a relation. Such causality can only be discerned if a relation changes over time. Consequently, the relative position describes the distance between two components, a motion equation describes how this distance changes over time, while an event in an instance of time describes how that pattern of motion changes. The occurrence of such event comes from outside of the considered system of two components, even if that “outside” means effectively within the physical boundary of one of these objects (e.g. a pilot changing the flight course of a plane, which would otherwise collide with another plane). Thus, only relations that involve temporal consequences have a direction indicating the propagation of induced temporal change and transfer of matter or energy from one component to another, which can be seen as another way of separating descriptive and performative relations.

The very existence of non-embodied entities is related to performative relationships. As described earlier, non-embodied entities emerge out of complex processes among very high numbers of embodied entities. Thus, a non-embodied entity can be considered to be an abstraction of a certain kind of performative relations. Using the building regulation example from the previous section, in consideration to the process of erection of many diverse buildings in a city, the regulation of e.g. maximum building height can be seen as a performative relation between the city and located in it buildings. Buildings can “grow” in height but if they are located in the city they cannot exceed the maximum height, where the city is in fact a complex aggregate of people and their habitats.

The implication of performative relations in architecture is twofold. Traditionally architectural structures are built to resist temporarily changing physical relationships, such as those between building structure and forces coming from using it people, machines, effects of changing natural phenomena, etc. Thus performative relations in that view on architecture may only be seen as performance in “dampening” of external forces. In the situation of external forces being greater than the integral “dampening” of the building system, change is inflicted on that system and the system is considered to have “failed”. On the other hand, from the perspective of adaptive architecture, the performative relations are the relations that are required for a system to adapt, since the building system is in a state in continuous motion and resulting transformation.

c) Transformative

The third kind of physical relations occurs when the physical relation (through some force) results in a process of transformation of the involved objects. This transformation means not only a change of property, thus internal or external organisation. It fundamentally implies alteration of internal matter and energy levels and their distribution, hence a flow of matter and/or energy in or out of the system. In that case, the relation has a direction indicating the flow or matter and/or energy from one entity to another. Such transfer causes transformation in both involved entities.

It could be argued that performative relations equally imply transformation of involved elements, since existence of force and its transfer implies a flow of energy, thus also transformation of these entities. Thus, similarly to how the performative relations can be seen an extension of descriptive relations between entities through inclusion of time, the transformative relations can be considered to be extensions of performative relations by taking into account that every performative process involves an inherent transformation of affected components, their continuous "becoming" of something else.

Because of such continuous "becoming" of entities connected by transformative relations, these relations are difficult to clearly describe. A convenient method for this is by doing this through a series of transactions, where a transaction always involves an exchange of some form of matter and energy. Non-embodied entities can serve as intermediaries in such transactions.

To refer to the mentioned earlier example of a building regulation, a building height regulation can be seen as a relation between a particular house and all the other houses in the city as well as with the city, being an aggregate system including these houses. Through its multiplication the same building regulation can also be seen as an autonomous, yet non-embodied entity. Enforcement of a building height regulation on a building exceeding the specified height (e.g. illegalay erected) results in the transformation of that building through its reduction in height or even demolition, while it also results in the transformation of the surrounding houses (e.g. by reducing the excessive shadows or allowing better views) and thus affecting their aggregate into the neighbourhood and a city.

1.4. Outlining aggregations

Examples of architectural system components and complex nature of relations that can be drawn between them provide basic insights into the difficulties in analysis and formation of such systems. It has been shown that architectural system components are in fact aggregates of other interrelated components. The traditional approach to systems is hierarchical in terms of rigid structure given to systems and sub systems in a single hierarchical taxonomy of parts and sub-parts, while, as argued by Christopher Alexander, in reality architectural systems exhibit a semi-lattice structure, where one component can be part of many systems. This section will dwell into characteristics of architectural semi-lattices and attempt to build foundations for productive ways of dealing with such systems without going into another extreme of subdividing these systems to an extent that would make them not incomprehensible to human cognition and perception, even when aided by most advanced instruments.

a) Systems as views on reality

Clearly, any phenomenon observed in reality can be endlessly broken down into components and subcomponents, between which endless relations can be drawn. Such phenomenon can also be endlessly extended, beyond any directly observable context.

When reality is being described as a system, that system has to be seen as representation of that reality and hence, also obviously a reduction of that reality. In this manner a system can be considered to be a "view" on reality, which view can differ based on its purpose and context.

b) Setting boundaries of a system and its components

A system is always defined by its boundary. The boundary separates what is in the system and what is outside of it.

Ay system can be defined as an *open system when it exchanges matter, energy*[1] or information (depending on the field it applies to) with its environment through its boundary. If we consider a building system as consisting of static, interconnected building parts only, it is a typical closed system. However, if we acknowledge existence of building components which respond to external factors, such building system may be qualified as open in aspects of information exchange or, in some cases, also energy, yet not in terms of exchange of matter. If we broaden the definition of a building system to include abstract building spaces and their users, or/and take into account that building elements can be added or removed during building operation, such building system can be regarded as an open system in all its aspects. All systems in reality are open systems, which exchange matter and energy through their boundary. Systems whose boundary is not permeable are closed systems and such systems don't exist in reality.

Each component of a system can be considered to be a system in itself. Thus also each system component has a boundary, which separates its sub-components from other system components and their sub-components.

Boundaries of components can be drawn in many ways which leads to system components that can overlap each other. One approach is to treat such overlapping components as parts of overlapping systems, which provide distinct views on one reality. Another approach is to integrate those systems by defining a new system where sub-components of original system's components are primary components themselves, while former components are groups of the new components. Similarly, if the external boundary of two systems doesn't overlap the external boundary of a system can be shifted to include components of both integrated systems. The disadvantage of integrating systems is the resulting complexity and illegibility this process leads to upon repetition.

c) Defining systems while tracing actor-networks

In Actor-Network Theory Bruno Latour proposes that reality is to be traced by mapping networks of interrelated entities following an assumed starting point of origin (depending on the object of study) and tracing relationships to other (living and non-living) entities, seen by Latuor as "actors". This strategy does not contradict systems view on architecture and resonates well in developments in interactive architecture, where buildings are often referred to as "networks for living in"[2]. Systems can be seen as a specific kinds of "panoramas", which in Latour's approach "provide the only occasion to see the 'whole story' as a whole. Their totalizing views should not be despised as an act of professional megalomania, but they should be added, like everything else, to the multiplicity of sites we want to deploy. Far from being the place where everything happens, as in their director's dreams, they are local sites to be added as so many new places dotting the flattened landscape we try to map. But even after such a downsizing, their role may become central since they allow spectators, listeners, and readers to be equipped with a desire for wholeness and centrality."[3] In this Latour also firmly suggests to "flatten" the studied system, by removing all hierarchies involved between

[1] Ludwig von Bertalanffy, 'The Theory of Open Systems in Physics and Biology', *Science* 111, no. 2872 (1950): 23–29.

[2] Mahesh Senagala, 'Building Is a Network for Living in: Toward New Architectures', in Proceedings of the 2005 Annual Conference of the Association for Computer Aided Design In Architecture (presented at the Smart Architecture: Integration of Digital and Building Technologies, Savannah (Georgia), 2005), 36–47.

[3] Bruno Latour, *Reassembling the Social: An Introduction to Actor-Network-Theory* (Oxford University Press, USA, 2007).

components and their sub components, thus when relations going “into” the boundary of a component become relevant, the sub-component these relations are pointing to should be treated as a component on the same “flat” level of hierarchy.

1.5. Understanding processes in systems

Once the components of an architectural system are defined, along with the boundary of the system and relations between its components, the way in which architectural systems operate as wholes can be brought into the picture. The following points are an attempt to concisely outline the key phenomena of the holistic behaviour of architectural systems emerging from local actions and consequently interactions between system components.

a) Actions-reactions

A system, by definition, requires action to take place among its components; otherwise it is only a set. Actions always involve a reaction. An action of an entity is always executed in relation to another entity or a group of entity. Because of this an action can be described through a performative or more extensively transformative relation. Chains of actions-reactions propagate through the system, building up its global behaviour.

b) Feedback loops

The recurrent cycle of action-reaction has been defined by Norbert Weiner as a feedback loop. A feedback loop occurs when the reaction to a given action causes change in the original action. A positive feedback loop enhances its cause while a negative feedback loop decreases it. A feedback loop is the main “building block” of regulatory systems, allowing the behaviour of such system, through negative feedback, to converge into a steady state.

c) Interactions

The process of interaction can be seen as a phenomenon which is an extension of the notion of a feedback loop. In an interactive process, every action is related to one or more preceding it actions and reactions of other entities to that action

It defines the action as something that is formed not by one entity but by two or more of them, in the process of action-reaction.

d) Causality, paths of development and entropy

The aggregation of feedback loops and consequently of interactions lead to the transformation of the entire system. The system develops over some path. In complex systems initial conditions play a role, although depending on system properties, also variations in initial conditions can lead to same stable states, either on local or global scales, otherwise referred to as attractors. Chaotic systems typically develop from high to low entropy levels (which is a measure of system’s non-uniformity and thus ability to perform work by itself).

e) Emergence

Through the development of a system, out of a multitude of local interactions and through recurring patterns, qualities and properties of the system may emerge that could not be anticipated from the study of its individual components and relations. Emergent properties could not be deduced from studying system components or their smaller assemblies in isolation. Examples of emergent properties range from the material properties such as liquid or solid state, through cloud formation to social behaviour of ants and people[1].

[1] Steven Johnson, *Emergence: The Connected Lives of Ants, Brains, Cities, and Software* (Scribner, 2002).

f) Stochastic processes

Emergent properties are outcomes of stochastic processes. Stochastic processes are characterised by consistent probabilities of their outcomes under same initial and external conditions. Following the developments of quantum theories, every process in nature can be considered to be stochastic, as every process can be broken down to an ultra-high number of quantum particles interacting with each other[1]. However, occurrence of attractors – steady states in such greatly complex systems results in very high probabilities given to a limited number of outcomes.

g) Robustness, stability and transformation

Attractors are an outcome of stabilising feedback loops. Systems without external interference tend to either large entropy distributions of components where energy and matter is spread evenly through the entire system, or can become 'caught' in an attractor, which produces a stable state with high entropy. Atoms and galaxies can be considered such steady states. A system which exhibits such steady state may be characterised by robustness. Robustness is a quality of the system to maintain its steady state under perturbations or unpredicted external interferences, thus to maintain its low entropy level.

h) Tendencies and capacities

Each system can be characterised by its tendency. A tendency of a system describes the direction in which a system is developing (e.g. towards or away from a steady state) in a given set of conditions. The capacity of a system is a broader term. It describes how a system would behave as a whole under a range of different conditions.

1.6. Complex processes in architecture

It has been shown that architecture can be treated as a complex system. It is a matter of convention how the boundary of an architectural system is drawn. Traditionally an architectural system would consist only of material building components and their assemblies, depending on the scale of concerned. It has been postulated that humans, other living organisms, other kinds of material objects, as well as non-embodied entities can also be included in architectural systems. It has been concluded that a boundary grouping system components based on their proximity and/or a boundary drawn based on the density of relations between system components is more appropriate than a any boundary drawn based on a predefined taxonomy. Such approach can allow for a more complete and thorough understanding of architecture as a complex system, its interactions and emergent nature of its properties and can lead to new ways of dealing with architecture, including broader possibilities for understanding and working with architectural adaptation.

2. Multi-agency of architecture

In its general sense, the term 'agency' denotes the ability of an entity to act in the world. Historically, in Western culture, agency has been considered to be an exclusive trait reserved to humans only and resulting from human free will – innate property of human essence. Contemporarily, while the idea of "essences" is commonly disregarded, presence of agency

[1] Han Feng, 'Quantum Architecture', in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012), 341–359.

has become equally acknowledged in animals and other living organisms. Eventually, agency can also be attributed to non-living material entities and, going even one step further; to non-embodied entities.

The phenomenon of agency can be seen as a specific kind of an emergent property of any system component (or a system as a whole, which as discussed earlier can itself be a component in another system), “agency is emergent in the brute sense of being unknowable in advance of specific performances”[1]. Thus, agency and its ways become only apparent when the given entity is put under a certain set of conditions leading that entity to undertake some kind of action. Such entity exhibiting agency is called an agent. The behaviour of an agent may on one hand be non-deterministic, but on the other hand it may be rational as well. Such rationality may however be highly individual and may largely depend on the history of the agent-components, their present and past relations to other components and their path of development.

The following sections are an attempt to provide an understanding of the phenomenon of agency in relation to its emergence and the role it has in architectural systems, which can consequently serve as foundation for further investigations into systemic aspects of autonomous architectural adaptation.

2.1. Agency and agents

In the words of Bruno Latour “anything that does modify a state of affairs by making a difference is an actor -or, if it has no figuration yet [i.e. is non-embodied], an actant. Thus, the questions to ask about any agent are simply the following: Does it make a difference in the course of some other agent’s action or not? Is there some trial that allows someone to detect this difference?”[2] The criterion proposed by Latour for agency is not the “intentionality” or the “meaningfulness” of the executed action, but the ability to create a lasting effect on the world. Consequently, we can recognize not only the agency of humans and other living beings, but also “the agency of the things that *produce* (helpful, harmful) effects in human and other bodies”.[3]

Looking at our world through the lens of such omnipresence of agency leads to the ontological position of considering the entire universe to be an infinitely complex system of interacting agents (where the boundary lies in infinity and where the system can be broken down and aggregated to an infinite number of components and where to *interact* means to communicate both ways - to exchange information and to mutually transform each other throughout the exchange process.) From such a broad perspective a network of densely interrelated with each other entities belonging to an architectural system can be considered to be a minute sub-system of the great multi-agent system of the universe[4]. In such a sub-system agencies of its constituent entities and interactions between them are intensified. This can be achieved by distinctive identification of objects and by opening up new, technologically augmented, channels for the exchange of information and mutual transformation of objects, thereby enhancing the natural ‘vibrancy’ of objects with new possibilities of sensing, processing of information and acting. Historically, these channels are related to the attribution of meaning

[1] Andrew Pickering, ‘Material Culture and the Dance of Agency’, in *The Oxford Handbook of Material Culture Studies* (Oxford University Press, 2010).

[2] Latour, *Reassembling the Social*.

[3] Jane Bennett, *Vibrant Matter: A Political Ecology of Things* (Duke University Press Books, 2009).

[4] Albert-Laszlo Barabasi, *Linked: How Everything Is Connected to Everything Else and What It Means* (Plume, 2003).

to things and their resulting symbolic properties. Contemporarily, new media technologies, embedded electronics, ubiquitous computing and other technology-driven cultural trends provide new alternatives for enhancement of architectural agency[1].

If we follow Latour's Actor-Network Theory, we can find agency in any type and any part of architecture. Once humans erect and begin to inhabit buildings (or any other architectural spaces), these buildings simultaneously begin to have a lasting effect on humans. This leads to humans modifying their habitats further, erecting new and different buildings, transforming, demolishing and re-erecting existing ones, continuously re-organizing and re-decorating, resulting in re-attributing meanings and functions to architectural places. This process can be observed in its full intensity in what we commonly label as 'primitive' architecture. Architectures such as nomadic settlements or slum cities are vibrant systems in the process of perpetual transformation, directly influenced by the local actions of inhabitants of these systems. Andrew Pickering refers to such processes as "dances of agency", "in which activity and passivity on both sides are reciprocally intertwined"[2]. On the other hand, in contemporary, industrialized architecture (from vernacular to modernist) such 'dances of agency' between inhabitants and architecture have become significantly slower processes. Paradoxically, with advancing building technology and strict, centralized building regulations opportunities for inhabitants to transform buildings have been greatly limited. "Architecture came to be seen as the conscious art of creating massive and perdurable structures, and came to see itself professionally as no more than that art, which is one of the reasons for their present problems and uncertainties."[3]

Consequently, it may be concluded that any kind of an architectural component, as they were outlined in section 1.2, can be treated as an agent. This implies that any living, non-living embodied and non-embodied entity may be considered to be the source of action, meaning that it may cause a lasting transformation of other entities – components of an architectural system, as well as entities beyond that system's boundaries.

2.2. Autonomy

The concept of agency is directly related to the notion of autonomy. A differentiation can be made between intrinsic and extrinsic sources of an entity's actions. To give an example, agency of a human being is normally considered to be a product of interactions between brain neurons intrinsic to that particular human. An agency of a chair that this human decides to sit on is extrinsic to that chair, as the chair "makes" the human sit on it not through the processes occurring within the boundary of a "chair system", but because of the human's awareness of chair's affordance and suggestion implied in that awareness that the chair is expected to be sat on. However, upon closer investigation such rigid separation between intrinsic and extrinsic origins of agency may be problematic. The neural system of human brain cells is formed through a long history of interactions with the outside world, while the internal structure of the chair, the way in which its parts are interconnected, is what gives it a particular affordance. Consequently, the intrinsic and extrinsic nature of observed agency is entirely dependent on the assumed boundary and temporal scale of the studied entity.

[1] Tomasz Jaskiewicz, 'Architecture as a Multi-Agent System', in *Volume#28, Internet of Things* (Archis, 2011), 56–60.

[2] Pickering, 'Material Culture and the Dance of Agency'.

[3] Banham, *The Architecture of the Well-Tempered Environment*.

It can be stated that in order for an entity to be considered as autonomous, the dominant source of its agency needs to be intrinsic to that entity. Otherwise, where the source of agency is extrinsic, autonomy becomes replaced by control. This definition of autonomy is a consequence of the earlier assumed understanding of a system. The same entity, depending on the way of its systemic “framing“ can be classified as autonomous or controlled.

Commonly architectural matter is considered as non-autonomous, and thus fully controlled by humans. In the traditional, essentialist, viewpoint humans intentionally erect buildings and cities. However, obviously those buildings and cities also shape the lives of humans forming a slow “dance of agency”. Yet, in such context the agency of architectural material components is constrained, as it only occurs extrinsically to them. There is little unexpected transformation coming from material architectural agents.

The concept of autonomy of a component in the context of this work will be used to describe the degree and frequency of such unexpected transformation of a given agent with a defined boundary. In respect to this, autonomy of an agent is proportional to the frequency of transformations induced by an agent, its indeterminacy in the system and consequently the ratio between intrinsic and extrinsic source of agency. The autonomy cannot be quantified in an absolute manner, but it can be defined comparatively among agents.

2.3. Communication

Communication is an activity of conveying information, where information is not only a set of data, but also involves interpretation of that data according to some convention. The fundamental model of communication has been formulated by Shannon and Weaver[1] and consists of a sender, message, transmission, noise, channel, reception and receiver.

In section 1.3 an overview of relations in systems was given. Yet, this discussion on relations cannot be complete without inclusion of communication as a specific kind of relation that is a consequence of the emergent agency of system components. While a single message can be compared to a descriptive relation, a communication process and related to it interaction between entities is a broader view and requires a transformative relation in order to be described externally.

Any transformative relation can be seen as a communication process consisting of at least two reciprocally sent messages, meaning that symmetric kind of transformation has occurred in both communicating entities, providing that there exists a mutual understanding between the sender and receiver of the message. That understanding, requiring an ability of interpretation is directly tied to the idea of agency as it needs to be performed autonomously and implies an ability of that entity to act on its own behalf and to induce transformation on its environment.

Traditionally, buildings are seen as mediums for messages conveying culturally important meanings. The meaning of such messages conveyed by buildings would be encoded by those buildings’ makers and decoded by their users. In the understanding of architectural communication which can be derived from Latour’s actor-network theory and consideration for buildings and their parts to be agents in their own right, buildings should be seen not as mediums, but as mediators. In this case both makers and users of architecture maintain continuous and bidirectional communication with buildings.

[1] Claude E. Shannon and Warren Weaver, *The Mathematical Theory of Communication* (University of Illinois Press, 1971).

2.4. Multi-agent systems

Multi-agent systems (MAS) consist of a number of autonomous agents that can communicate with each other[1,2]. In this way multi-agent systems don't require imposing a hierarchy on their components. This allows for a very high degree of flexibility and reconfigurability. Within such systems multiple levels of feedback loops may also be embedded. This means that they can remain in a constant process of adaptation, thus being often nondeterministic (stochastic), as opposed to parametric systems where each set of parameters fully determines the state of the entire system and can produce only one output.

The primary element of a MAS is an "intelligent agent". Intelligent agents are autonomous entities that observe and act upon their environment and direct their activities towards achieving certain goals. MAS are commonly applied in various fields, including scientific simulations of complex phenomena, generation of video special effects, computer games and many others[3].

Agent based models is an emerging methodology that allows us to explore complex system that are too hard to define and explore analytically. As such, the idea is to define many simple entities (agents) with simple behavioural rules, combine them in a multi-agent environment that is also governed by simple rules forming an ecosystem of interacting devices[4], and observe as their local interactions cause the emergent of some global impact on the overall system (such as the emergent of some equilibrium point).[5]

2.5. Interactive architecture as a complex adaptive system

The definitiont of a complex adaptive system (CAS) is close than this of a multi-agent system. "CAS are without exception made up of large numbers of active elements that are diverse in both form and capability"[6]. However, what sets complex adaptive systems apart is adaptation, which occurs in individual elements, as well as consequently in aggregates of such elements and in the system as a whole. In CAS, every component can be treated as an agent and at least some of these agents are learning agents. Behaviour and/or structure of agents can thus transform over time as the system develops. Even though the behaviour of each of the system's agents may be driven by relatively simple, linear and rational logics, the entire system as a whole is likely to exhibit highly intricate and non-deterministic behaviour. Complex adaptive systems are capable of dealing with not predefined situations

[1] Yoav Shoham and Kevin Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations* (Cambridge University Press, 2008).

[2] Michael Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. (Wiley, 2009).

[3] M. Wooldridge and W. van der Hoek, 'Multi-Agent Systems', in Handbook of Knowledge Representation, ed. Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter (Elsevier Science, 2008), 887–928.

[4] S. Coradeschi and A. Saffiotti, 'Symbiotic Robotic Systems: Humans, Robots, and Smart Environments', *Intelligent Systems, IEEE* 21, no. 3 (2006): 82–84.

[5] Nigel Gilbert, *Agent-Based Models*, annotated edition (Sage Publications, Inc, 2007).

[6] Holland, *Hidden Order*.

and adapting to unforeseen conditions. They can also be easily expanded and altered by increasing the number of elements or by changing behaviours of all, or of a selection of its contained parts. [1, 2, 3]

2.6. Architecture out of control[4]

The result of attribution of agency, autonomy and ability to communicate to non-living architectural components opens a new perspective on dealing with architectural systems. Rather than following Patrick Shumaher's interpretation[5] of Niklas Luhmann's theories[6], which lead to an understanding that architecture is a by-product of inter-human communication, Bruno Latour can be followed instead to reach an understanding of architecture as a heterogeneous network that is being continuously formed among human and non-human, material and non-embodied agents. Such network is boundary-less, however its aspects can be outlined through systemic models where the most relevant components and relations between them are defined and around which a boundary is drawn (Latour refers to such models as "panoramas").

As a result, feedback loops between human and non-human architectural entities naturally become the fundamental issue of concern for architecture. Aggregations of such entities and using methods already successfully applied in other domains of concern for systemic enquiry. Architecture becomes seen inherently as a system in perpetual motion and development, where larger scale systemic phenomena emerge out of local interactions between components.

Consequently, instead of understanding architecture to be a process of human control of their habitats, architecture needs to be seen as being inherently out-of-control, as a process which can be set in motion and modulated through local interferences and adjustments, but not determined in respect to its global outcomes. As a result "space-customisation, (...) is one of the modes of emergence and self-organisation interactive buildings can be based on"[7].

2.7. Architectural evo-devo

The other consequence of, both virtual and physical, employment of such dynamic, multidimensional architectural design matter is the necessity to take into account how such architectural systems operate over time, among others, to make sure that its emergent qualities don't develop threats to its users. Two key terms borrowed from biology can facilitate description of such temporal qualities of a complex system. These terms are *development* and *evolution*. Development refers to changes in a system of a single organism over its lifetime. Evolution means changes in the system of entire species consisting of numerous individual

[1] Ang Yang, *Intelligent Complex Adaptive Systems*, 1st ed. (IGI Publishing, 2008).

[2] John H. Miller and Scott E. Page, *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*, illustrated edition (Princeton University Press, 2007).

[3] Yin Shan, *Applications of Complex Adaptive Systems*, 1st ed. (IGI Publishing, 2008).

[4] Kevin Kelly, *Out of Control: The New Biology of Machines, Social Systems, & the Economic World* (Basic Books, 1995).

[5] Patrik S. Schumacher, *The Autopoiesis of Architecture: A New Framework for Architecture*, 1st ed. (Wiley, 2011).

[6] Niklas Luhmann, *Social Systems*, 1st ed. (Stanford University Press, 1996).

[7] Henriette Bier, Kathleen de Bodt, and Jerry Galle, 'SC: Prototypes for Interactive Architecture', in *Proceedings of the 12th International Conference on Interactive Technologies and Sociotechnical Systems*, VSMM'06 (Berlin, Heidelberg: Springer-Verlag, 2006), 21–28.

organisms. The field of evolutionary developmental biology, in short evo-devo, studies the connection between development and evolution of organisms, mostly on the molecular DNA level[1].

The mechanisms explored by evolutionary developmental biology interconnecting genetic triggers and to developmental processes of organisms[2] can be a very useful parallel for constructing non-biological complex systems. Just like development of a living organism is closely linked to evolution of its entire species, development of a complex adaptive building system can be influencing evolution of the entire system framework. In this way, first experiments may be limited in their performance, yet, gradually, further iterations of system evolution will lead to a mature framework applicable to real world projects. What's more, with use of digital software tools, multiple developmental scenarios can be performed virtually, allowing for significant system evolution even within a singular developmental cycle of an architectural construct.

3. Designing autonomy

The consideration for architecture to be a complex system built up of human and non-human components and the consequent attribution of agency to all such entities provides new opportunities for approaching architectural adaptation. Understanding architecture as a system implies existence of some behaviour of the system where relations between its components continuously change. Such "systemic" attitude towards architecture has been gaining popularity throughout last decades and has been interpreted and dealt with in a diversity of ways (as discussed in section 1.). Approaching architecture as a *complex adaptive system* means that a high number of individual components of a given architectural system are additionally capable of autonomous behaviour and adaptation. Resulting complex architectural adaptation involves a non-predefined path of development of architectural systems and requires consideration for occurrence of emergent architectural qualities and phenomena (as discussed in section 2). This approach requires a new way of designing architecture. The currently widespread architectural design methods are aimed at creation of architecture which is in a steady state, rather than being a system in non-predetermined motion. As Gordon Pask has phrased it "the role of the architect here is not so much to design a building or city as to catalyse them; to act that they may evolve"[3]. In older to develop a constructive critique of commonplace architectural design processes, a broader look needs to be taken at the nature of design in general, and more specifically, at the common architectural design practice along with its traditional tools and instruments. Following such critical overview a constructive approach towards finding new methods for dealing with complex adaptive architecture can be taken. Pro-active role of new technologies, as among others vastly explored by Saggio[4], needs to be consequently related to these developments.

[1] Sean B. Carroll, *Endless Forms Most Beautiful: The New Science of Evo Devo and the Making of the Animal Kingdom*, 1st ed. (W. W. Norton & Company, 2005).

[2] H. Allen Orr, 'Turned on, a Revolution in the Field of Evolution?', *The New Yorker*, October 24, 2005.

[3] Frazer, *An Evolutionary Architecture*.

[4] Antonino Saggio, *The IT Revolution in Architecture, Thoughts on a Paradigm Shift* (Lulu Press, 2010).

3.1. Designing as solving moving problems and embracing opportunities

In its broadest sense, an act of designing involves a creative process of finding one or more solutions to a set of one or more problems or taking advantage of an opportunity for change, otherwise defined as "initiation of change in man-made things"[1]. "New needs grow and old needs decay in response to the changing pattern of facilities available. To design is no longer to increase the stability of the man-made world: it is to alter, for good or ill, things that determine the course of its development."[2] Typically, problems requiring creative design solutions have wide solution spaces (many possible solutions) and they frequently lack unambiguous problem definition and solution selection criteria. In consequence, design processes evade deferring to rigid procedures. Often, formulation of design output is attributed to the "creative genius" of a designer and "good" design follow's designer's broader agenda. Nevertheless, many attempts have been made to rationalise the design process and to find ways of increasing its efficiency.

Design methods are "any procedures, techniques, aids or 'tools' for designing"[3]. In the seminal work on design methods, John Christopher Jones defines three fundamental stages of any design process to which design methods can apply; divergence, transformation and convergence. Divergence is defined as "the act of extending the boundary of a design situation so as to have a large enough, and fruitful enough, search space in which to seek a solution". Transformation involves "pattern-making, fun, flashes of insight, changes of set", which "can occur unexpectedly at any time, but (...) should only be applied after sufficient divergence has occurred". At the third stage of convergence "the problem has been defined, the variables have been identified and the objectives have been agreed. The designer's aim (...) [is to] reduce the secondary uncertainties progressively until only one of many possible alternative designs is left"[4].

Nigel Cross follows on Jones's ideas by saying that "methods of design might be viewed as representing three different perceptions of design activity: that of creativity, that of rationality, and that of control over the design process. 'Each of these three views of designing' (...) 'can be symbolized in a cybernetic picture of the designer. From the creative viewpoint the designer is a black b*ox* out of which comes the mysterious creative leap; from the rational viewpoint the designer is a *glass box* inside which can be discerned a completely explicable rational process; from the control viewpoint the designer is a *self-organising system* capable of finding shortcuts across unknown territory.' (...) Viewing the designer as [neither a magical black box or a fully rational glass box, but] a self-organising system means adopting a viewpoint that recognises the full intelligence of the designer, and yet also recognises that intelligent behaviour is self-reflective and capable of improvement, and can benefit from tuition and from using some forms of external aids."[5] Cross also underlines the importance of designer's intuition as a key factor of success in any design process. Lawson defines a design process similarly, replacing divergence, transformation and convergence by parallel analysis, synthesis and evaluation seen as a way of negotiation between design problem and solution[6]. As expressed in those

[1] John Chris Jones, *Design Methods*, 2nd ed. (Wiley, 1992).

[2] Ibid.

[3] Nigel Cross, *Engineering Design Methods: Strategies for Product Design*, 4th ed. (Wiley, 2008).

[4] Jones, *Design Methods*.

[5] Nigel Cross, 'The method in their madness; Understanding how designers think' (Delft, the Netherlands, 1996).

[6] Bryan Lawson, *How Designers Think, Fourth Edition: The Design Process Demystified*, 4th ed. (Architectural Press, 2005).

models, the process of design is rarely a linear one, “problems and solutions in design are closely interwoven”[1]. Definition of problems may change and evolve alongside construction of design solutions.

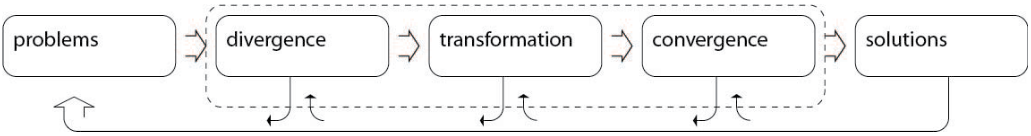


Fig.15. General design process, following J.C. Jones

The most traditional, “creative-genius” or “black-box” kind of designing is a solitary activity. Reflecting back on the origins of the “design methods” movement, Jones argues that the crux of “new” design methods is the fact that “the new methods permit collaboration before ‘the concept’, the organising idea, the back-of-the-envelope-sketch, ‘the design’, has emerged (...). The new methods, properly used, release everyone from the tyranny of imposed ideas and enable each to contribute to, and to act upon, the best that everyone is capable of imagining and doing.”[2]

Design methods depend on the subject of design. Additionally, based on different design philosophies, different aspects of design can be prioritised during a design process more than others, leading to varying design approaches, such as user-centred design, performance-driven design, use-centred design, agile (action-centric) design, artistic design, experience design or many others, some of which favour more rational, while other more unstructured methods. The combination of design topic and approach determines the set of methods to be used.

Design methods can be grouped based on the design process stage they apply to (although, especially in less rational approaches, these stages can be tightly interwoven). Methods such as contextual enquiry, trend spotting, surveys and observational techniques, or literature research are performed in the problem formulation phase. They also help in design divergence stage, where the main objective is to broaden the problem perspective. Both divergence and early transformation can be performed using brainstorming or synectic methods. Different kinds of models (from sketches to mock-ups and prototypes) can be used in transformation and convergence of design. Converged solutions can be evaluated using experiential prototyping, user evaluation or criteria ranking and weighting. Active involvement of users in all stages of a design process is referred to as participatory design, which topic has been raised in the architectural debate since the mid-20th century, but not reached the mainstream practice[3].

A distinction can be made between design methods and design principles. The first are procedures which are to be followed in order to reach a good quality design solution. The second are rules of good design practice that can, but do not have to be part of any specific method. Principles of diverse character such as TMTOWTDI (there is more than one way to do it), Ockham’s razor (parsimony – choose simpler solutions above the more complicated ones), golden section proportions or modularity[4] provide a wide variety of design guidelines, without imposing any particular design procedure.

[1] Jones, *Design Methods*.

[2] Ibid.

[3] Peter Blundell-Jones, Doina Petrescu, and Jeremy Till, *Architecture and Participation* (Taylor & Francis, 2005).

[4] William Lidwell, Kritina Holden, and Jill Butler, *Universal Principles of Design* (Rockport Publishers, 2003).

An important aspect in any design process is the role of design models; the "need to use sketches, drawings and models of all kinds as a way of exploring problem and solution together, and of making some progress when faced with the complexity of design"[1]. A model is a simplified representation of a system or phenomenon. In design context the term model usually refers to a simplified representation of reality, either existing or proposed. The types of models range from mental models, through conceptual models, sketches, diagrams, plans and sections, virtual models to physical scale models or full scale mock-ups. In the process of design simple models become transformed into more complex ones to eventually result in the actual design product, gradually reducing its degree of abstraction and simplification. Models are often strongly related to particular design methods. Computer aided design (CAD) software introduced a new kind of tools to crate design models, which deliver unprecedented means of assisting designers with handling design problems of much higher complexities.

The aftermath of the "design methods" movement seems much greater in the fields of industrial design and engineering, computer science or systems design than in architecture. In these domains highly specific design methods are frequently applied with much success (although the "creative-genius" designing always remains to some degree complementary to the systematic use of design methods). In architecture the role and application of structured design methods is much less significant than in other engineering disciplines. The "creative genius" / "black box" view on the architect's design process still dominates and rational design methods are often considered as limitation to architect's individualist expression and creativity, and are seen as contradictory to the artistic, ambiguously defined symbolic, historical and cultural role of architectural design. Nevertheless, also in architecture certain commonly recognised design methods can be outlined.

3.2. Architectural design

Although architectural design could be seen as a subset of a much more general domain of design, architecture has a longer tradition as a separately recognised design discipline and therefore does not share common practices with other, younger design domains. Most generally architectural design involves finding solutions to complex problems of spatial organisation, in answer to intricate societal, cultural, technological and environmental needs and beliefs of architecture's inhabitants. In order to understand the challenges faced by design of complex architectural systems, a broader look at design methods is necessary.

Architectural designing is part of a conventionally linear process of briefing-designing-building-operating and occupying[2] of architecture. Traditionally, design is subdivided into sequential stages. It starts by interpretation of the design brief by architect formulating a general concept that provides an outline of how the project is expected to answer demands stated in the brief. Gradually more detail is added to the design. Subsequently, other parties involved in the design, engineering and construction of the project sequentially contribute to it with their expertise, adding more detail to the design. In this process, consequent steps frequently require revision of the earlier ones, resulting in multiple iterations of the entire design process, or some of its parts. The design process typically ends with preparation of building plans and is followed by building construction, occupancy and ultimately redesign or destruction of a building.

Early stage architectural design methods usually involve discussion of the brief with the clients and conceptual brainstorming of the design team, aiding the design problem divergence. Nevertheless, most of the early design work is performed in an unstructured creative process.

[1] Cross, 'The method in their madness; Understanding how designers think'.

[2] Foque, *Building Knowledge in Architecture*.

Those design aspects involve creative interpretation of the design brief into design problems and finding an original solution to those problems. Very often challenged problems have little relationship to the functional aspect of architecture and may involve its cultural, symbolic, philosophical, artistic or aesthetic qualities. The final outcome of an architectural design process, is an integration of solutions to all accounted for design challenges, encapsulated in an architectural form.

The most common contemporarily design method encountered already in early phases of architectural design is functional programming[1]. Functional programming involves the definition of functions to be included in the project and organisation of these functions in relation to each other and in the three dimensional space of the project site. Distinctive functional patterns and relations between them are used as rudimentary building blocks of the architectural program. These patterns are most commonly taken as-is, based on established precedents and are often deeply rooted in culture, norms and legislation.

Typological architectural design methods may additionally complement the initial design phase and functional programming. These methods are based on using established building types as reference points for designs, adapting architectural patterns found in existing buildings to suit the given brief and concept. A building type involves a particular program, its spatial organisation, but also encompasses a specific building form, building techniques, materials and even decorative elements. Many architects define their own typologies or reject the concept of architectural type altogether, seeing it as an unnecessary design constraint. Creative architectural designers take more freedom in reinterpreting the functions and finding alternatives solutions for how particular functionalities of building systems can be achieved through architectural means, thus going beyond standards and conventions of common functional patterns and creating new typologies.

Throughout the design process, architects heavily rely on the use of various models. The most simple model in architecture is a sketch. Architectural design often starts with a series of sketches, trying to encapsulate the essence of a particular architectural concept. These sketches are subsequently transformed into plan and section drawings and more accurate perspectives. Next to design drawings, physical 3d scale models are commonly built. Recently much of such models are being created using computer tools. To illustrate more abstract concepts including the functional program, also diagrams are commonly employed throughout the entire design process. Models are also deeply rooted in architectural theory[2] and the process of working with the model may be considered to be an interaction between the model and the designer, where through a “dance of agency” the designer creates a model, and the model provides new insights for the designer to come up with changes to the design, thus a modification or creation of a next version of the model, until the satisfactory design solution is found.

Common approaches to architectural design have encounter broad critique. “While function is still regarded as a major issue in architecture by almost all its proponents, its understanding is ad hoc and intuitive. (…) Enquiry into functional innovation is directed towards the more technologically orientated aspects of the discipline, notably in the fields of low-energy design and materials development. Few high profile architects have shown any consistent attention to these areas in their design, paying lip-service at most. The empirical and traditionalist attitude to function in architectural design is perpetuated, even where the scope of the problem is not effectively addressed by these means. (…) Architectural theory is regarded as the driving force in (architectural) creative design, but in general, this theory is derived from disciplines outside its field, and tends to the philosophical and esoteric. Thus,

[1] Anthony Vidler, ‘Toward a Theory of the Architectural Program’, *October* - (October 1, 2003): 59–74.

[2] Healy Patrick, *The Model and Its Architecture* (010 Uitgeverij, 2008).

‘creativity’ is measured by an often formalist response to a verbal discourse. This has been promoted in all the visual arts through the development of their own critical elite, each with its own self-referential discourse. Increasingly throughout the twentieth century this has isolated lay people from any debate through its alienating use of language“[1]. This condition of architectural design may be partially attributed to disillusionment in systematic design approach following many of its failed attempts in modernist architecture. Due to the high complexity of factors that architectural design has to deal with, modernist architectural solutions based on reductive analysis of problems led to projects, which although succeeding in addressing some of the problems, were at the same time overlooking other problems and generating many unexpected and highly undesirable conditions. Due to the complexity of encountered problems qualitative judgments appear more successful than quantifiable ones.

These problems became exaggerated and fuelled much architectural conservatism, such as Robert Krier, who writes: “A mere half century ago modernist movements claimed to have developed definitive solutions to all the problems of the built environment. Today, one truth is evident: without traditional landscapes, cities would be a nightmare on a global scale. Modernism represents a negation of all that makes architecture useful: no roofs, no load-bearing walls, no columns, no arches, no vertical windows, no streets, no squares, no privacy, no grandeur, no decoration, no craftsmen, no history, no tradition. Surely the next step must be to negate these negations.”[2] Clearly, there must be an alternative to Krier’s approach of negating modern architecture and reverting to backward traditionalism.

In the modernist paradigm, an architectural design would have been considered as successful when, after being built, it would function in accordance with its originally intended use. Additionally, satisfaction of other, easily measurable criteria, such as the cost of building and its maintenance, could also be taken into account along with the subjective view of the building as artistic expression of the architect (with satisfaction of end users being often dismissed in their importance). Yet, such definition of architectural design’s success becomes problematic when architecture is considered to be a system of many entities and a continuous process, not a singular and static object. In many cases, originally intended function may dramatically change over time and yet, the architectural space can be regarded as successful, even though it may then serve a different purpose it can be potentially bringing greater benefits to its owners and users. Thus the reconsideration of the evaluation criteria for architecture may be the first of steps to take on the path towards a new methodology for complex adaptive architecture.

In view of this, satisfaction of inhabitants can be postulated as the main criterion for architecture’s “success”. Depending on the context of the project, the term “users” may refer to occupants or inhabitants of the space, but also to stakeholders of the project, who although don’t need to directly participate in the given architectural space, their decisions and evaluation is critical to that space’s operation.

What’s noteworthy, architecture typically needs to benefit a large number of users, and hardly ever isolated individuals. In this, potential conflicts can naturally emerge, since what is considered as good for one individual may not be good for another, generating conflicts, not only between individuals, but also between entire social groups (as can be seen with the recent public debate about introduction of a mosque next to the WTC memorial site). It is the ethical role of architecture to not only minimize such conflicts, but also through spatial intervention to induce a transformation of the society that would make these conflicts obsolete.

[1] Anne Fitchett, ‘Function and Creativity in Architecture’, *Urban Forum* 9, no. 2 (June 1998): 271–278.

[2] Leon Krier, *The Architecture of Community*, 1st ed. (Island Press, 2009).

In this way, architecture can be seen as an active constituent of the societal ecosystem. As Jane Jacobs remarks: "Vital cities have marvellous innate abilities for understanding, communicating, contriving and inventing what is required to combat their difficulties". "Living" qualities of space can be achieved in a cycle of continuous, re-design and adjustment of cities on all scale levels, in response to societal and environmental changes, but also pro-actively inducing the transformation of architecture's content and context.

From this understanding of architecture and architectural space stems the logical conclusion that an ability of space to adapt to changing external and internal conditions is its primary quality. Architectural spaces that can be frequently transformed to match changes in the conditions of their internal and external, artificial and natural environment are bound to be more "successful" than those spaces that do not adapt to these changes.

In conclusion, in order to embrace complex architectural adaptability, not only the design process needs to be reasserted in order to permit for design of non-deterministic human habitats in place of architectural stable-states. Also the criteria for evaluation of architectural designs need to be altered. The satisfaction of participants of architecture and the continuous development of the holistic performance of architectural systems need to be given priority. In order to gain more insights to possible strategies for implementing such changes, a closer look needs to be given to design tools and instruments, which as presented in the following argument, play critical roles in how the design processes are carried out and implemented.

3.3. Design instruments

Traditionally, architects are limited to providing in their designs only a very limited number of singular variants for each project at most, with no more than few scenarios of how their designs could change over time. Creating buildings as systems means delivering wide and flexible ranges of design solutions depending on multiple parameters of diverse kinds, derived from various phenomena such as environmental conditions or socio-spatial ecologies.

a) Evolution of design instrumentarium

Ever since antiquity, building designs have been produced and communicated by means of two-dimensional drawings. Ancient Egyptians used papyri or flat slabs of limestone to sketch and draw plans and views of buildings to be erected. This technique remains in many of its aspects unchanged until today. It is still the case that the most common way of documenting architectural projects is by means of representing designs using flat plans, sections and elevations. Development of geometrical techniques such as perspective of axonometry allowed using two-dimensional media to represent three-dimensional spaces in more appealing ways than plans and sections. Until relatively recent days the only way to create spatial designs in real three dimensions was with the use of scale models, often accompanying the two-dimensional drawings. However, the developments in computer aided design have begun to radically change ways in which architecture can be designed.

First computer applications that were created to support design, operated based on ideas directly mimicking traditional ways of drafting using hand drawing boards. Their development can be traced back to Ivan Sutherland's Sketchpad system[1] which used a handheld pen-like device as an interface to a computer program generating simple line drawings appearing on a display screen. What originally was the domain solely belonging to computer graphics (CG), over time branched into a more specialized field of computer aided design (CAD). Along increasing proliferation of computers, CAD have greatly facilitated drafting processes by

[1] Ivan Sutherland, 'Sketchpad: A Man-machine Graphical Communication System', *Technical Report* no. 574 (1963 2003), http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-574.pdf.

allowing for digital storage of drawings, easy editing, copying of elements, fast measurements, scaling, plotting etc.. Gradually more features were introduced to CAD applications, allowing for automation of generic and repetitive actions, such as creation of documentation, executing predefined sequences of commands and ultimately associative and parametric drawing. Nevertheless, the overall method of two dimensional drafting remained almost the same as with drawing on paper.

What made CAD software a truly revolutionary tool for designing, was the one innovation entirely unprecedented in traditional media; the development of three dimensional (solid) modelling. Instead of creating a series of flat representations of one design, CAD systems gradually introduced a possibility of creating three dimensional models of designed products. Two dimensional representations, being sections or plans, perspective or axonometric views can then be automatically generated from that one model. This allowed for complete integration of what formerly were two different ways of working; two dimensional drawings and three dimensional models. Furthermore, digital 3d models allowed application of various computerised engineering simulations, integrating CAD with computer based engineering (CAE). They also led to appearance of parametric and associative modelling techniques, where instead of directly changing the model's geometry; certain features could be indirectly modifiable using intermediate parameter values[1].

2d and 3d models being stored digitally can also be directly translated to code for CNC (computer numerically controlled) machines using computer aided manufacturing (CAM) software to directly produce accurate physical elements. In parallel to its application in design and engineering, 3d computer models became also widely used in film and computer entertainment industry, where diverse solutions have been used, allowing 3d animation and other rich visual representations

Computer tools provide novel means to rigorously approach the complexity of architectural design. However, most of commonly encountered architectural design software tools enforce the established, linear design process model, as it has been traditionally enforced in architectural praxis. Even though dealing with architectural designs as systems is not a conscious decision for most practicing architects, contemporary digital design tools inherently treat virtual design models as systems of interrelated components. In simplest cases such as Autcoad, these components are geometric primitives, in more specialised software, including BIM (building information modelling) solutions, every component of a virtual design model is tagged with highly specific information, is strictly classified and has established relations to other virtual components (for example in Gehry Technologies Digital Project or Autodesk Revit). Changes to one component's properties may thus involve consequent transformation of all other related to it components of the model. In all cases virtual models are restricted to representation of material building components and are destined to be static or at most changing only within a predefined range after the building's completion.

Design software is normally being employed to facilitate tasks complimentary in the project and usually corresponds to different specialists' domains. Because data produced by each of these applications is often required by other applications to conduct their part of the design and engineering process, apart for distinct functionalities of these applications it becomes an equally important issue how various software applications can communicate between each other.

Traditionally, projects are executed in cycles. The general building design, engineering and construction cycle consists of, schematic design, design development, construction documents, bidding, construction, occupancy[2], with each of the steps often leading to

[1] Benjamin Aranda and Chris Lasch, *Pamphlet Architecture 27: Tooling*, 1st ed. (Princeton Architectural Press, 2005).

[2] http://www.cod.edu/facilities_plan/DESIGN1.pdf

a result requiring going back one or more steps and repeating parts of the process until satisfactory results are achieved. Going beyond the scale of a single project, this cycle may be seen as a continuous loop of program, design, execution, evaluation phases, including potential sub-loops on most time-scales. In more generic terms this cycle consists of: aim, tool, execution and judgement phases[1]. Regardless of the scale or scope of a project, at each stage an evaluation can take place and certain parts of the process may be repeated in order to achieve a better result.

Normally project cycles are executed in steps, where different steps correspond to different actors participating in the process. Nevertheless, definition of these steps may vary. Steps can also include sub-steps and it's essential to remark that evaluation takes place throughout the entire process, with certain more explicit evaluation moments. For example, in a usual architectural design and construction process first a client would formulate a demand for a project, then an architect would define a conceptual design, followed by a detailed one, subsequently installation engineers would perform structural detailing and add technical layers to building plans and ultimately the building can be constructed and inhabited or otherwise occupied. The cycle may include multiple loops, for example between client and architect or architect and building engineer in order to re-evaluate previously taken steps when problems are encountered.

Existence of feedback loops in such processes in itself implies interactions between actors in the process. However, the communication process is normally very slow and limited. The logical consequence of accelerating this process could have an effect of accelerating the process and/or increasing the quality of achieved results by allowing for many more process iterations within the same amount of time. To give an example, a form of a building is passed on to the building structural designer who in turn sends proposed structural solution to a structural engineer to perform structural calculations. Traditionally each drafting film (before 1950s drafting linen or drafting cloth) was used to draw additional layers over basic design plans and sections. Use of computer applications allows to create such layers within CAD software (in two and three dimensions), and to exchange data digitally instead of using printed copies. In this way no quality is lost in the process and the actual act of exchange information can last seconds, even when large distances are involved. In order to allow this kind of exchange, all parties need to use same standards for data they exchange.

b) BIM

Connectivity between software models used on different stages of design and engineering processes can accelerate communication between design and engineering process actors. There are two approaches to facilitate this. One is to exchange only information relevant at the particular moment in the design and engineering process. To this day most file standards used are proprietary and make exchange of information between software applications very limited. Often data is being lost in this process or errors are created. However more commonly software applications move away from proprietary file formats and allow open standards, for example based on human readable XML (extensive markup language) notation.

The other approach is to create an integrated model to which all involved specialists can read and write data. In building design the term BIM (Building Information Model) has been introduced for this and has become the leading trend in facilitating exchange of information among building design process and lifecycle participants. The main purpose of BIM is thus to bring together all information constituting the project in one model, potentially stored in one file or database. However, most BIM solutions to this date remain proprietary, although their common compliance to the Industry Foundation Classes (IFC) developed by

[1] E.D. Huisbergem and I. Kriens, 'Plan Cycle', in *The Essence of Spatial Planning and Strategy* (TU Delft, 2003).

BuildingSMART Alliance[1] and open source projects such as BIMserver[2] ensures that in near future connectivity between various applications through Building Information Models will be much less constrained.

c) Simulations

A special kind of tools utilised in architectural design are simulations. The role of the simulation is not so much to provide a representation of the model, but to approximate its performance. Typically considered performances are structural, energy use and traffic and routing. In all such cases the simulation is deployed to provide an insight into the way in which a building or any other architectural construct will operate once realised. While a model created to represent space depends on designer's agency in the process of design evaluation and modification, the simulation's outcome has its inherent agency. It is deployed in order to point the designers to the direction which is unknown before simulation's deployment.

d) Design tool interfaces

Examples presented above show a variety of solutions in which designers may employ computation to aid them in design, engineering and prototyping tasks, including design and prototyping of interactive systems. In order to use these tools a designer has to be able to communicate with the computer. This communication happens through a human computer interface[3].

Commonly an interface consists of a screen used to send information from computer to user and keyboard and mouse (or an alternative pointing device such as trackball, tablet or touch sensitive layer over the screen) that are used to send information from user to computer. Communication between user and computer may occur multiple cycles per second. Information communicated on the screen usually includes direct feedback of latest actions of users, such as updating the position of a cursor based on the pointing device displacement or showing letters as they are being typed on the keyboard.

A keyboard is a perfect device for input of text. Common pointing devices such as a mice or pen tablets are very suitable for two dimensional interaction with information. However they do have limitations for CAD applications. Firstly 3d navigation may be difficult and counter intuitive. Secondly, these interfaces are not suited for situations when more then one user is involved. For this reason many alternative interfaces are being researched and some already applied to CAD. Following main categories can be defined. Computer to user communication (3d displays, augmented reality displays, multiscreen and surround displays), user to computer communication (specialized handheld controllers, motion tracking, speech and voice recognition), bidirectional user-computer communication (multi-user touch surfaces, computer supported group design and decision making spaces).

e) Post-design processes

As a result of employment of those new technological means, we can already witness a radical shift in architectural design methodology. Linear and centrally steered "top-down" designing becomes gradually replaced or supplemented by creation of parametric, procedural and relational digital design models. Consequently, those models become final design products themselves, making traditional, fixed plan drawings less important or even completely redundant. They also allow an intensification of the "dance of agency" between the designers and the model in a design process. Consequently, the lifespan of the initial virtual model

[1] http://www.buildingsmart.com/bim

[2] http://www.bimserver.org/

[3] Brenda Laurel, *The Art of Human-Computer Interface Design* (Addison-Wesley Professional, 1990).

becomes extended beyond the conceptual design phase. Through BIM database (e.g. defined using IFC-industry foundation classes), a model can be used by other design experts in the design process, to perform design engineering, among others of the structural performance, climate, environmental performance, energy usage, circulation and many others, allowing them to feedback directly to the initial conceptual design decision.

The digitization of the design process has its effect also in next following stages of building development. There, mass production can be replaced by mass customization of building elements, made possible with data-driven manufacturing technologies. In this way, in many cases, it does not make a significant difference whether a building consists of repetitive components, or if its every single component is a different variation[1].

Ultimately, after being built, buildings can be equipped with sophisticated building management systems (bms) that dynamically govern HVAC installations, lighting, security, accessibility, media systems, and many other aspects of building's functionality. Yet, building automation of today remains detached from architectural design. Building installations are commonly being added to buildings after the architectural design is considered finished. What's more, dynamic features of buildings are used to respond to predefined situations only, therefore allowing building's flexibility, but not adaptation. Rare experiments show that alternative building management systems can be built[2], but no mainstream solutions exist.

This shows that although they are hardly interactive, contemporary buildings are already commonly designed, constructed and used as dynamic systems, however structure of these systems does not allow any degree of indeterminacy, thus interactive qualities cannot emerge. What's more, the systemic nature of created virtual design models and their consequent transformation into physical human habitats is hidden from most designers under constraining design software interfaces.

f) Design instruments

The distinction between a “tool” and an “instrument” in their understanding posed throughout this dissertation may appear subtle. However that distinction is critical to the role that design instruments are to have in the proposed design approach. In its general sense, “tools” extend abilities of humans by providing “prosthetic” enhancements of human bodies with the purpose of fulfilling specific tasks in a specific, narrowly defined way, and these tasks only. A hammer is thus a tool, which allows concentrating the human force with the purpose of driving a nail, any other use of a hammer is considered to be its misuse. “Instruments” similarly to “tools” may enhance human abilities by providing specific affordances and allowing distinct interactions. However, the end goal and exact specificity of their operation is not predefined. In this way, although there are defined techniques for playing musical instruments, the kind of music they are used to play and the interpretation of music to be played is open and unfolds throughout the interaction between the musician and the instrument. Consequently, the music is produced through the interaction between the artist and the instrument. Arguably any tool can be in fact used as an instrument, and any instrument could be reduced to a tool. Therefore the distinction is not in any material qualities that a tool or an instrument possesses, but in the way they are being employed.

Traditional way of drawing building plans involves the use of tools. A drafting pen and transparent paper were tools that architects have been employing before the digital era to document their drawings, but rarely to design. On the other hand, perspective drawings can

[1] Stephen Kieran et al., *Refabricating Architecture: How Manufacturing Methodologies Are Poised to Transform Building Construction*, 1st ed. (McGraw-Hill Professional, 2003).

[2] Peter Anders and Michael Phillips, ‘Arch-OS: An Operating System for Buildings’, 2004.

be treated as instruments. Introduction of perspective has changed the way in which buildings have become designed on the break point between medieval and renaissance architecture and drawing in perspective has been used to refine and explore design concepts.

The initial role of digital CAD tools was to improve the speed and efficiency of traditional drafting tools, by providing extensions to human abilities to document a design, create a final model, but they did not aim to assist or influence the creative design process itself. Nevertheless, the use of digital CAD tools allowed increased number of iterations over design alternatives and project adjustments, facilitated by automation of detailing adjustment, engineering calculations and design documentation. Parametric and procedural design models were originally introduced only in later design stages to facilitate design synthesis. Recently they have begun to play a role also in the conceptual (design convergence and transformation) design stage. During the first decade of the XXI century, creative applications of parametric and procedural design models have widely proliferated and diversified and the trend continues (see e.g. SmartGeometry workshop and conference series).

Traditionally in architecture different models created in the design process use different media. Extensive use of digital technologies allowed introducing an integrated design model for all stages of the design. Building Information Models (BIM) can be used throughout all design stages and which can be accessed and modified by all design participants. Building information modelling becomes commonly accepted by building industry and it is widely recognized for its great potential to transform the architectural profession; “we need to acknowledge the implications of the massive expansion of data and move on from a performative analytical model to a more comprehensive conceptualisation of information modelling that opens up creative options leading to new qualities and relationships, and does not just streamline a process. It should expand the ways we use data rather than merely generating taxonomies or collecting an envelope of constraints. Some of the best designs have been those that have broken the rules and gone beyond technical optimisations or the prescribed constraints of clients and municipalities. Rather than limiting our choices, information modelling can open us up to the new way of thinking and its massive potential.”[1] An integrated design model permits a tightly integrated design process, where conceptual design and design engineering can be performed simultaneously and can reciprocally influence each other.

Consequently, what originally was just a tool, has become an instrument. The abilities of digital platforms in creation of virtual parametric associations, simulation, sharing of design data, precision, modification and embedding of procedural logic in virtual models has led a growing number of architectural designers to explore new design possibilities and has led to many new design methods and new criteria for assessing created designs. Most of such designs, however, are still considered to be explorations of new possibilities and they rarely address the questions of adaptation in architectural systems.

The difference between contemporary digital tools and instruments lies mostly in the interface. The interface of a CAD environment hides the underlying logics of the model, its components and relations between them. On the other hand, working with digital modelling and simulation software by directly scripting involves generating performative design processes with full understanding of the underlying systemic complexity of created model. This approach allows creating relationships, feedback loops and eventually emergent behaviours in virtually formed systems. As in recent years has been demonstrated by the Rhinoceros 3D Grasshopper community, in this way geometric modelling software can be used to build models that go far beyond what the software creators might have anticipated.

[1] Cynthia Ottchen, ‘The Future of Information Modelling and the End of Theory: Less Is Limited, More Is Different’, *Architectural Design* 79, no. 2 (2009): 22–27.

The employment of instruments, not only as “prosthetic” devices to their users, but also as prosthetic devices to space, has influenced architecture throughout centuries. “The challenge is how architecture, ‘as means of concretisation’ (…) can consciously take on this new informal dimension”[1].

3.4. Towards designing complex adaptive architectural systems

Described developments in digital architectural design modelling show a new trend in creation of architectural models, where working from “within” the model’s logic allows for development of new qualities and an intense “dance of agency” between a designer and a model. However, applications of models created in this way are limited to the design process only. The role of the model ends with creation of the building plans and consequent erection of static building structures. The dynamism of created models is not utilised to develop building adaptations.

It is a logical next step to “not freeze the process” when the erection of the building starts[2]. The architectural models developed as representations of future habitats could be developed in such a way, that their inherent logics are employed to drive the adaptive processes in actual architectural spaces. To achieve this evaluation criteria of modelled processes have to become attached to changing factors. For example if the model’s evaluation criterion is the structural performance of the building in correlation with material costs, the model develops into a static outcome. On the other hand, if the model is evaluated based on earlier postulated user satisfaction, it is certain that such satisfaction will be changing in ways unpredictable to model designers and will be partly independent of the given state of the design. Thus, feeding back real-performance data into procedural architectural model can be the first step towards architectural adaptation.

The approach of opening the model to an external input raises many questions. How should such model deal with locality of inputs and their multiplicity?

a) Applying multi-agent systems

Multi agent systems are commonly applied to simulations, where they function as models of real-world phenomena. Finite element structural analysis, crowd simulations, energy and wind flow simulations are all based on creation of a very high number of autonomous agents in space with relatively simple, local behaviours. Through their local interactions, real life phenomena can be approximately predicted, that would not be possible through linear equations.

The creation of an entire architectural design as a whole as a multi-agent model allows for local adaptations in such system and provides a promising framework for creating a bridge between digital design models, modelling adaptive operation of architecture and eventually delivering actual architectural systems capable of meaningful adaptation in real-world situations. The application of multi-agent systems to architectural adaptation is yet highly unexplored and will become a critical part of investigation in further presented research. Yet, in order to provide the complete picture, and before these research investigation can be brought to their application, development in from which architectural design can benefit will be brought into view.

[1] Antonino Saggio, ‘The Search for an Information Space’, ed. Kas Oosterhuis and Lukas Feireiss (presented at the Game Set and Match II, Delft, the Netherlands: Episode Publishers, Rotterdam, 2006), 214–219.

[2] Kas Oosterhuis, *New Kind of Building Non-standard Arch: A Designers Guide for Non-Standard Architecture* (NAi Publishers, 2011).

b) Learning from systems engineering

Approaching architecture as a dynamic system is new for architecture. However, systems engineering is an established discipline, which is by its definition interdisciplinary and from which architects can gain insights into systems they design. "The development of large systems in which many different disciplines participate requires holistic lines of thinking. This means that the requirements and structures of a system are looked at totally detached from the knowledge of specific details."[1] "Systems engineering is a multidisciplinary approach to transform a set of user needs into a balanced system solution that meets those needs. Systems engineering is a key practice to address complex and often technologically challenging problems. The process includes activities to establish top-level goals that a system must support, specify system requirements, evaluate alternative system designs, evaluate the alternatives, allocate requirements to the components, integrate the components into the system, and verify that the system requirements are satisfied. It also includes essential planning and control processes needed to manage a technical effort."[2] "Systems engineering integrates all disciplines and describes a structured development process, from the concept to the production to the operation phase and finally to putting the system out of operation. It looks at both technical and economic aspects to develop a system that meets the users' needs. "[3]

Numerous strategies have been applied in systems engineering. Waterfall engineering strategies, which follow a highly rigid and uncreative procedure became replaced by methods such as Prototyping, Spiral, Agile software development, Extreme Programming, Joint application design, Lean software development, Rapid application development, Scrum and others, often specific to the type of systems being designs. A language SySML (Systems Modelling Language) has been developed as an extension of a subset of UML (limited to software engineering) in order to provide shared and transparent methods for analysing, describing and engineering systems, regardless of their specifics. The language includes diagramming conventions for modelling system aspects such as requirements, block definitions, parametric relations, internal blocks, activity, use cases, sequences and states. SysML or other modelling conventions alongside diverse systems engineering strategies can be applied to the engineering of architectural systems. However, their rigidity may not be appreciated by creative design professionals. Yet, most importantly, the system testing which in systems engineering occurs before the final deployment of the system may be impossible in architectural applications.

c) Architecture and software

"Architecture becomes a game and the users the players. Architects are the programmers of that game."[4] Development of computer software may thus be considered as a useful analogy to the production of buildings. Lowest-level electronics can be programmed using low level programming languages such as assembly, but on top of assembly, powerful languages such as C++ or Java had to be introduced in order to allow software engineers to more swiftly create computer applications. Such applications may include even more simplified programming languages such as Vbasic or Python to allow their users to enhance these applications' functionality, whereas most of the end-users are happy to limit their work with the application to typical, standard and easy to use graphical interface. Similarly, interactive

[1] Tim Weilkiens, *Systems Engineering with SysML/UML: Modeling, Analysis, Design* (Morgan Kaufmann, 2008).

[2] Sanford Friedenthal, Alan Moore, and Rick Steiner, *A Practical Guide to SysML: The Systems Modeling Language*, Revised (Morgan Kaufmann, 2009).

[3] Weilkiens, *Systems Engineering with SysML/UML*.

[4] Kas Oosterhuis, *Architecture Goes Wild: Manifest Writings* (010 Uitgeverij, 2001).

architecture is likely to evolve in a similar way, with lowest level standards for their operation being firm and least frequently revised, higher level elements being creatively explored by specialists (new architects) and users being able to participate in some of this creation on the highest level. The current problem is that there is no appropriate low-level framework for interactive architecture that would allow higher level creative explorations. Current systems developed for creation of smart environments are typically proprietary and limited to specific medium. The new promising development in this area is the umbrella technology of the Internet of Things[1]. Yet, until Internet of Things standards become established, a more provisional solution need to be used to facilitate the parallel development of higher-level prototypes of interactive architecture.

d) Learning from Interaction design (IxD)

In product design the importance of interaction between the product and its users has been given substantially more research attention than in architecture. Although the term "interaction" can be applied to describe relation between a user and a traditional (passive) kind of product, however the new dimension of the term concerns interactions with products which are digitally enhance, thus which can sense, process and affect back their users in an active manner. As Bill Moggridge writes, "The information revolution has changed the way we interact with everything, from the games that we play and the appliances in our homes to the tools that we use at work. Some interactions are designed so that we don't notice them (…). Other interactions are designed so that we are very conscious of them."[2] Contemporary interaction design involves screen-based experiences, interactive products, and services. Experiential prototyping in all three categories is essential to their success. "Prototype early and often, making each iterative step a little more realistic. (…) You will only know that the design is good when you have tried it out with people who will use it and found out that they are pleased, excited, motivated and satisfied with the result" [3] urges Moggridge. Formulation of design concepts followed by prototyping needs to go in pair with user research[4] and testing. Prominence of user research in product design and engineering is therefore understandable. Experience design, and more specifically user-experience design have thus become strong fields in research and industrial design praxis.

Simple prototyping can be performed using mock-ups and techniques such as "wizard of Oz", where a hidden person operates the interactive device. Beyond the early proof-of-concept mock-ups, actual logics of the device need to be programmed and tested. For this a computer can be used. Software platforms such as Adobe Director, Adobe Flash, MAX MSP, 3DVIA VIRTOOLS or open source programming environments such as Processing or OpenFrameworks provide tools for designers to rapidly create deploy behaviours on a personal computer. These systems can use standard computer interfaces or can be connected to sensors and actuators, which can be embedded in product prototypes (e.g. using Phidgets or Arduino Frimata). Instead of using a personal computer, system logic can also be directly programmed into the microcontroller, such as Arduino or PIC, which come with designer oriented and easy to use programing platforms. Prototypes created in this way come close to actual interactive devices built on mass production scale. Due to decreasing costs of technology and open source platforms, cost of development of such prototypes has greatly decreased in the recent decade.

[1] Rob van Kranenburg, *Network Notebook #2: The Internet of Things*, 2007.

[2] Bill Moggridge, *Designing Interactions*, 1st ed. (The MIT Press, 2007).

[3] Ibid.

[4] Mike Kuniavsky, *Observing the User Experience: A Practitioner's Guide to User Research*, 1st ed. (Morgan Kaufmann, 2003).

Observing the “experiential prototyping” trend in design, leads to conclusion that design models become widely replaced by design prototypes. Designers no longer create representations of designed products, but actual functional entities, which can be deployed and tested in the real world early on in the design process. Although creation of prototypes of product prototypes has always been part of design on product scale, creation of interactive experiences for such product prototypes that can be deployed autonomously and can directly interact and evaluate user experiences is a new trend, and profession-revolutionising trend.

3.5. Hybridizing the design-use process

Discussed trends inevitably lead to the integration of design processes and processes of the actual operation of architecture in one, hybrid system. “Building information modelling makes it possible to link the briefing (or programming) phase and the design phase with the construction phase and the operation of a building into one comprehensive whole.(…) BIM technology will allow the architect to design and build in virtual space and investigate the consequences of his ideas and concepts in real time. This technology can lead to a concurrent, integrated and interactive designing-building process. With that vision there no longer exists a division between different phases of the project, as was the case in the traditional models. In fact, we should no longer speak about phases in the traditional sense but rather of project levels: the level of the brief (or program), the level of design, the level of construction, the level of building use.(…) The realization of a project can be seen as an evolutionary process, where data undergo a metamorphosis from abstract ideas to concrete facts and pass from a virtual universe into the real world.”[1]

Consequently, building adaptation can be seen as not a quality which is pre-designed in the architectural system, but which naturally emerges during that system’s operation. The operation may consequently mean the operation of the actual building, as well as of its design model. The design model, consequently can be rapidly transformed into the system it represents. Increasing role of experiential prototyping reduces the role of models in design processes. A reduction gradation can be postulated between reality, system and model, where the system is a reduced view of reality and a model is the representation of the system. In a situation where iterations between models and systems are frequent, the models can be seen as interfaces to systems they represent.

a) From model to framework

The role of the design model in the described new context diminishes. Design models are representations of to-be reality. Since in the hybridized design-use approach designers and participants together create actual architectural systems already from the early phases of their design activity, there is no need for intermediary models created before the actual systems are put in place. The idea of a framework can be put against the idea of a design model. A model implies a constrained and reduced view on existing or designed reality. A framework could be compared to a construction scaffolding. As opposed to a specific model, a framework is an open set of methods, instruments and conventions which can, but don’t have to be employed in order to construct an actual system. A framework, can be modified and adjusted to suit what is being built with it and it provides a set of generic and flexible blueprints for various degrees of system’s design and implementation.

In view of this, other kinds of models, may still be justified. The second kind of models employed in architecture are aimed at better understanding of occurring deployed phenomena and interventions into them. These models can be referred to as analysis models. Analysis models

[1] Foque, *Building Knowledge in Architecture*.

are essential for humans to grasp the complexity of phenomena that otherwise cannot be embraced by human cognitive capabilities. Analysis models can be seen as interfaces between complex systems and human cognition.

The third kind of models is simulations. Simulations can be considered to be extensions of the actual systems. Although their role is to simulate the real system, they are also performing as systems in their own right. The simulation model can thus be seen as a specific component in the actual system, which influences the development of that system.

The fourth kind of models that can be employed in creation of complex adaptive architectural systems are meta-models, thus models that represent sets of rules, constrains, methods, theories and frames in which a given architectural system unfolds in its entirety. In other words, a meta-model represents a framework in which a given system is created and which it uses to operate.

b) (non)standards

In systems engineering standards play an important role as they allow reuse of elements in a system. Here standards often imply patterns, methods or techniques that can be shared among systems, while system's operation can be fully unique. On the other hand, standards in architecture have bad connotations, since the reuse of architectural components means repetition of the same space and rejection of local uniqueness, hence also contradicts local adaptability. The development of standards in architecture implies stagnation, and design for no-change. Therefore, to avoid discrepancies, the word "convention" will be used throughout the rest of the dissertation to talk about standards in the sense of agreed upon patterns of operation (such as e.g. communication protocols), and the word "standards" will be used in respect to rules constraining the diversification, uniqueness and consequently adaptation (such as e.g. standard size norms), although in some indicated cases the separation between the two will be less obvious.

c) Open source

In common language and software jargon the term open when used on its own implies "open source", rather than an open system. The open source concept originates from computer science and means a system of which the structure and operation is not concealed, but open for modifications and improvements by anybody. If applied to creation of a building system, this concept implies that building users may not only have full access to the knowledge of how the building system operates, but can also modify and enhance it according to their specific needs (within agreed constraints).[1]

The idea of open source paired with the concept of experiential prototyping means that users of the system may not only learn about the system's operation through the experience of participation in interaction with the system, but they can also gain insights into the internal logics of the system and consequently change these logics. In this way, the distinction between the user and designer clearly dissolves. It also means that system's adaptation can be equally embedded in the system itself or can come from the user. Framing the problem differently, the user seen as an integral actor in the system can be the source of adaptation of another actor, by modifying it and by having an overview of the broader mechanisms of interactions among all system's actors.

[1] Moreno Muffatto, *Open Source: A Multidisciplinary Approach* (Imperial College Press, 2006).

3.6. Model, metamodel and framework

a) Model

In architecture, the term "model" traditionally refers to scaled representation of a building's form, which precedes creation of the actual building[1]. In context of the arguments presented to this point, such traditional role and understanding of architectural models contemporarily devaluates. In design research case studies that are to be traced in chapter IV, architecture is shown as developing continuously through a process, in which designers, experts and inhabitants interact with virtually prototyped and experientially deployed (actual) architectural systems. This interaction occurs directly, without implicit need to produce models as intermediary products of an architectural (building creation) process. However, once the traditional, and in this context devaluated, kind of architectural models becomes removed from the repertoire of architectural design instruments, a different, and, in the context of presented research, more appropriate, architectural interpretation of the notion of a model can be introduced.

In the broad context of systems engineering, the term *model* refers to "a physical, mathematical, or logical representation of a system entity, phenomenon, or process"[2]. In this understanding, models mainly provide abstraction of systems and thus are mainly aimed at representing their structure rather than state. Such representation of system structure may serve us to understand (analyse) existing systems as well as to build (synthesize) new systems.

Accordingly, models of system structure in architectural context represent thus not a singular and permanent spatial or functional composition, but an ontology and structure of relations between system components and their organization. Consequently, such models can represent architectural systems, regardless of particular states of these systems or temporal configurations. Clearly, such understanding of models stands in synchrony to approaches discussed further in chapter IV.

b) Metamodel

Following such "redirected" understanding of a term "model", the term "metamodel" can consequently be added to the architectural vocabulary. The notion of a "metamodel", has been adopted from software and systems modelling. Metamodel is an explicit model of the constructs and rules needed to build specific models within a domain of interest.[3] The concept of a metamodel is critical to development of interactive architecture. Where a model describes a general structure of an architectural system, a metamodel defines conventions in which a family of models is to be made to coherently describe different kinds of stable structures of one or many architectural systems.

c) Framework

The third notion fundamental to this chapter is "framework". The concept of a "framework" is broader and more loosely defined than a "metamodel". In its common sense the term "framework" refers to a basic conceptual structure delivering a certain specific view on reality. It describes the general ontological structure in which a given system operates and defines how particular tools and instruments can operate in such view.

[1] as discussed in chapter II - section 3.2

[2] John Leonard, *Systems Engineering Fundamentals* (DIANE Publishing, 1999).

[3] http://infogrid.org/wiki/Reference/PidcockArticle

The notion of a framework is used differently across domains. In qualitative research theory, the term framework generally refers to the notion of "conceptual framework", meaning "the system of concepts, assumptions, expectations, beliefs, and theories"[1] that "explains, either graphically or in narrative form, the main things to be studied—the key factors, concepts, or variables—and the presumed relationships among them"[2]. In architectural design research, the idea of the conceptual framework can be analogically used to describe the most general ontology of constituents of an architectural system and relations between them.

The other common use of the term "framework" can be found in software engineering. "Frameworks model a specific domain or an important aspect thereof. They represent the domain as an abstract design, consisting of abstract classes (or interfaces). The abstract design is more than a set of classes, because it defines how instances of the classes are allowed to collaborate with each other at runtime. Effectively, it acts as a skeleton, or a scaffolding, that determines how framework objects relate to each other."[3] Commonly, a "software framework" is thus understood as a collection of classes, libraries and APIs (application programming interfaces). A software framework provides generic software components reusable in a larger number of applications within a given domain.

In context of this research, the term framework denotes a different kind of framework: the "architectural design framework". In vernacular context, an architectural design framework can be seen as an accumulation of traditional building techniques based on repetition and craftsmanship. Such traditional architectural design framework is inherently tied to an essentialist worldview of buildings being considered as permanent spatial interventions Traditionally established building design techniques, tools, instruments, building and planning legislation and cultural conventions are all parts of such architectural design framework. The design research case studies traced in the previous chapter demonstrate that traditional architectural design frameworks become devaluated in the context of interactive architecture. There is thus need for a new framework for architectural design. Such framework is required to provide a replacement to traditionally established worldview of architecture being static, architectural design being based on repetition, functions of buildings being predetermined, and inhabitants not being considered to be parts of architecture.

Analogically to conceptual frameworks, such framework needs to define a general worldview for interactive architecture. However, it also needs to go beyond an expression of that worldview. Analogically to software frameworks, it is also required to provide a generic system structure which can become a basis for diverse architectural applications. It needs to define the fundamental elements and relations between them, out of which architectural systems are to be created, while not constraining any of the possible iA applications.

d) Integration

There is clear interdependence between a framework, a metamodel and a model. A metamodel can be a part of a framework as it defines rules of how architectural systems are to be modelled. An architectural system may be described by a number of models, it can consist of different agents, can be developed using diverse design instruments. The iA systems are in perpetual motion, their state continuously changes. A system model describes relationships among system components and system qualities that don't change in between system states. The model can also define the specific agents that constitute the system, their properties and behaviours. As observed in traced case studies, the model of the architectural

[1] Joseph A. Maxwell, *Qualitative Research Design: An Interactive Approach*, 2nd ed. (Sage Publications, Inc, 2004).

[2] Matthew B. Miles and A. Michael Huberman, *Qualitative Data Analysis: An Expanded Sourceboo*, 2nd ed. (Sage Publications, Inc, 1994). (quoted after Maxwell)

[3] Dirk Riehle, *Framework Design: A Role Modeling Approach* (ETH Zurich, 1999).

systems can undergo bifurcations, transforming from one stable state to another, such as e.g. upon construction of a physical prototype. In different stable states systems can exhibit different qualities and different agents can be identifiable within these systems. In that cases a system model would also change, as its principal rules of operation would be altered. In all cases the mata-model of general rules and constructs of changing models in itself remains constant.

A system is a reduction of reality. A model of the system is a reduction of all combined states of that system. A metamodel is a reduction of all system models developed in that metamodel and a collection of fundamental ontological properties shared by all models and systems. A framework is a combination of a meta-model, a convention for forming models and systems and instruments to aid in these tasks.

3.7. Existing metamodels

The specification of a framework is a complex undertaking. Historically, in architectural design and construction "frameworks" have developed as combinations of cultural conventions, craft and tools, and legal regulations. They function as such commonly to this day and vary locally.

However, the use of computer technologies in the building industry demands a more structured approach. Industry foundation classes (IFC) are a global undertaking to establish a universal metamodel for modelling building systems and are in the process of becoming the official International Standard ISO 16739. On the other hand, systems modelling languages such as UML or SysML provide generic methods for definition of metamodels facilitated by the meta-metamodel (MOF M3) which offers the definition of the main types of UML "building blocks" (MOF::Classes). In this way UML and SysML can remain generic and implementation independent. In common design practice even more abstract, simplified and ad-hoc metamodels are often used to conceptually model systems.

The question to be answered is to what extent a metamodel for iA can be specific and to what extent does it need to stay undefined to allow development of non-predetermined, complex adaptive interactive architectural systems? Can listed metamodels be adopted entirely or partly for this task, or is an entirely new approach needed?

a) IFC as a metamodel

As of 2011, industry foundation classes (IFC) provide the leading open standard for comprehensive modelling of complete conventional building systems. The IFC model data is organised in four layers. "The core layer provides the basic structure, the fundamental relationships and the common concepts for all further specializations in aspect specific models. (...) The shared element data schemas contain intermediate specializations of entities. Entities defined in this layer can be referenced and specialized by all entities in the domain specific layer. (...)The domain specific data schemas contain final specializations of entities. Entities defined in this layer are self-contained and cannot be referenced by any other layer. (...) The resource definition data schemas consist of supporting data structures. Entities and types defined in this layer can be referenced by all entities in the core layer, shared element layer, and the domain specific layer."[1]

[1] http://buildingsmart-tech.org/ifc/IFC2x4/rc2/html/index.htm

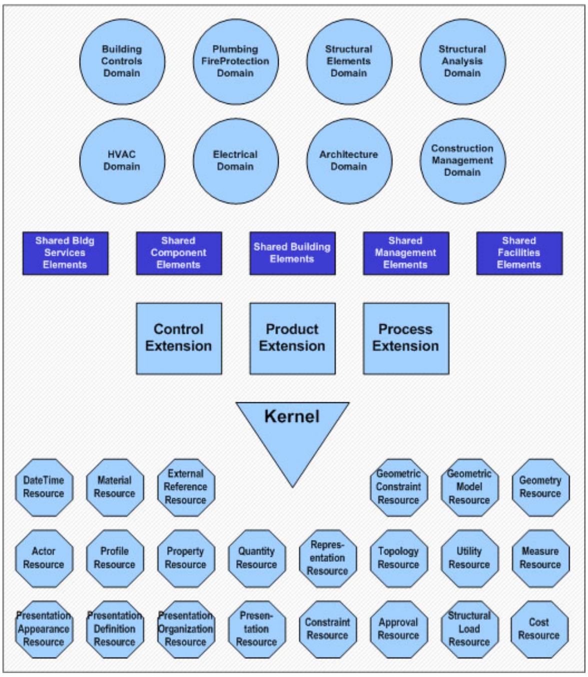


Fig.16. All IFC data schemas with shared element data highlighted (IFC2x4 release candidate 2)

b) Modelling languages as metamodels

The SysML and UML are modelling languages that are related to each other. Both SysML and UML can be used to build models and metamodels. A closed meta-metamodel, defined in Layer-3 of Meta-Object Facility(MOF M3), provides the self-referenced definition of a meta-class, by which all classes used in UML/SysML metamodels are inherited.

UML is a widely used language used to model object-oriented software. SysML extends and modifies UML to make it better applicable to model a greater variety of systems. For this reason SysML, although less ubiquitous, is more applicable to challenges of interactive architecture. A meta model such as IFC, as well as a specific project developed using IFC could both be modelled in UML or SysML. SysML would be however a better candidate for it, as it uses diagramming conventions more suitable for non-software exclusive systems. In SysML classes are replaced with blocks, defining basic system components.

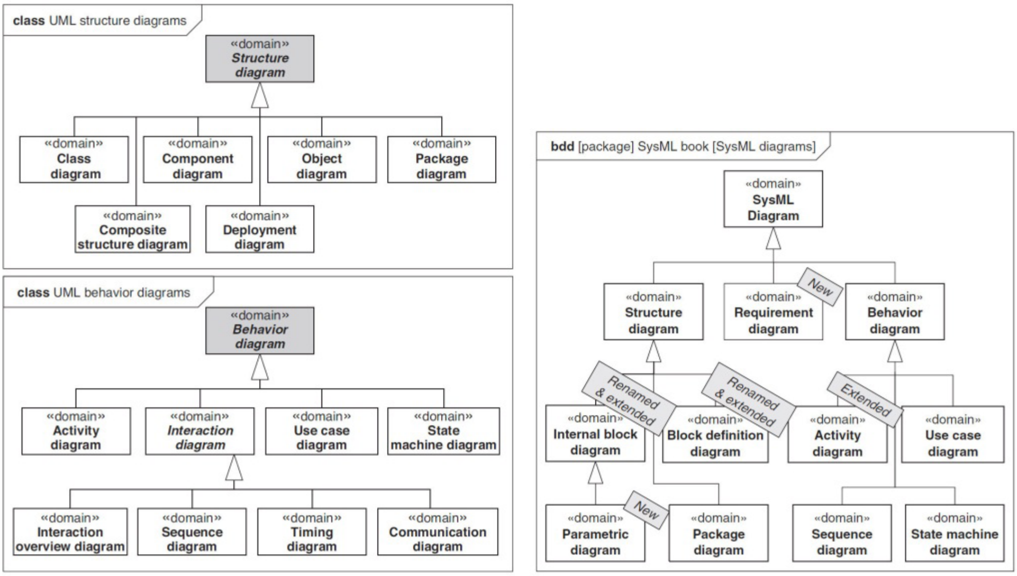

class UML structure diagrams
«domain» Structure diagram
«domain» Class diagram
«domain» Component diagram
«domain» Object diagram
«domain» Package diagram
«domain» Composite structure diagram
«domain» Deployment diagram
class UML behavior diagrams
«domain» Behavior diagram
«domain» Activity diagram
«domain» Interaction diagram
«domain» Use case diagram
«domain» State machine diagram
«domain» Interaction overview diagram
«domain» Sequence diagram
«domain» Timing diagram
«domain» Communication diagram
bdd [package] SysML book [SysML diagrams]
«domain» SysML Diagram
«domain» Structure diagram
«domain» Requirement diagram
New
«domain» Behavior diagram
Renamed & extended
Renamed & extended
Extended
«domain» Internal block diagram
«domain» Block definition diagram
«domain» Activity diagram
«domain» Use case diagram
New
«domain» Parametric diagram
«domain» Package diagram
«domain» Sequence diagram
«domain» State machine diagram

FIGURE 3-4

Overview of UML diagrams.

FIGURE 4-3

SysML diagrams.

Fig.17. Comparison of UML and SysML diagram structure[1]

c) Multi-agent metamodels

There are numerous existing metamodelling conventions, standards and tools that support and facilitate creation of (rational) agents. For creation of individual agents, Russel and Norvig provide a general overview of the existing ontological agent conventions; "Simple reflex agents respond directly to percepts, whereas model-based reflex agents maintain internal state to track aspects of the world that are not evident in the current percept. Goal-based agents act to achieve their goals, and utility-based agents try to maximize their own expected "happiness". All agents can improve their performance through learning."[2] Within that, BDI (Beliefs-Desires-Intentions)[3] agents are currently a prominent agent knowledge modelling paradigm. MAS are typically developed in software, but can also be found in robotic applications (e.g. Swarm-bots[4]) and are often associated with development of artificial intelligence[5] (e.g. swarm intelligence). MAS can consist of simple-reflex or learning (intelligent) agents.

Communication between agents may be seen as a separate modelling problem to the modelling of individual agents. Agent communication languages (ACLs; e.g. FIPA or KQML) define and standardise communication protocols between (software) agents. Cayci, Callaghan and Clarke discuss the need for a distinct communication language for Intelligent Buildings and present a prototype (DIBAL)[6].

[1] Weilkiens, *Systems Engineering with SysML/UML*.

[2] Stuart J. Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach* (Prentice Hall, 2009).

[3] Michael Georgeff et al., 'The Belief-Desire-Intention Model of Agency', in *Intelligent Agents V: Agents Theories, Architectures, and Languages*, 1999, 1–10.

[4] Francesco Mondada et al., 'Swarm-Bot: A New Distributed Robotic Concept', *Autonomous Robots* 17, no. 2 (2004): 193–221.

[5] Russell and Norvig, *Artificial Intelligence: A Modern Approach*.

[6] Filiz Cayci, Victor Callaghan, and Graham Clarke, 'A Distributed Intelligent Building Agent Language (DIBAL)' (presented at the 6h International Conference on Information Systems Analysis and Synthesis, Orlando, Florida, 2000).Florida.

Agent-oriented programming languages (e.g. AgentSpeak or GOAL) have been developed especially for the agent-based paradigm of programming. Software creation of multi-agent systems may be greatly facilitated by integrated Development Platforms (e.g. Jade with its own programming language and using FIPA-ACL for inter-agent communication, Jason for AgentSpeak language, Jack being a proprietary platform with FIPA compliance).

Computer games are a specific kind of popular and technologically advanced applications of multi-agent systems. Thus, many game engines and libraries (Such as Unreal Engine 3 or Ogre 3D) can be also employed to facilitate creation of software multi-agent systems. Other platforms (such as 3DVIA Virtools and Studio, Cycling '74 Max|MSP or Processing) provide simplified integrated development environments (IDE), which allow even non-programmers to create rich multi-agent applications. An notable experimental platform is Breve, which bears some resemblance to game engines and allows for development of multi agent system simulations in a virtual 3d world using python. Similarly, MASON provides an easy to use Java framework for creative multi-agent experiments.

d) Ad-hoc metamodels

During architectural or other design processes, ad-hoc diagrams and sketches are often used to model the meta-operation of the designed system. Rarely are these sketches done in a systematised manner; often designers develop their own sketching and diagramming techniques. Creative designers consistently defy adoption of methods such as UML/SysML to design. This shows that despite the generic nature of these methods, they don't provide enough flexibility to permit creative processes, nor intuitive usability.

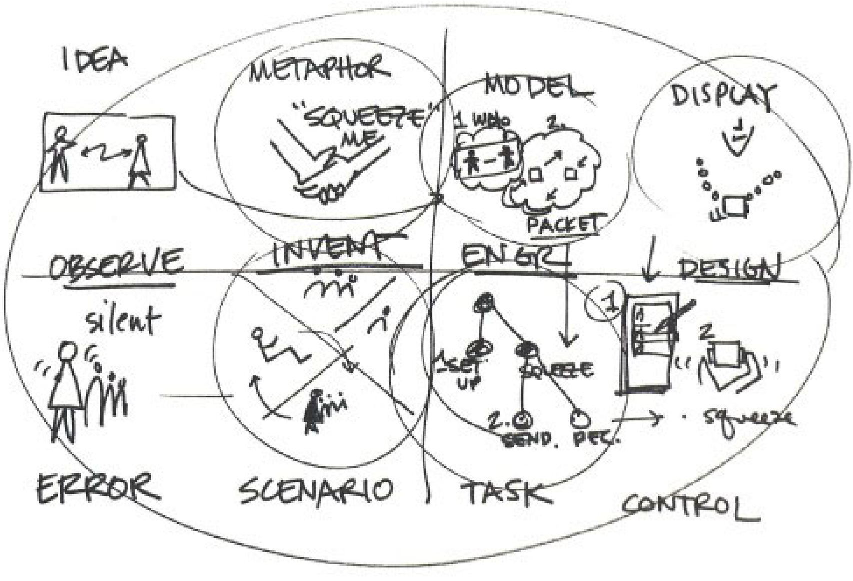


Fig.18. Sketched metamodel diagram[1]

e) Open metamodels

In design research case study projects to be discussed in chapter IV, reference to models has been avoided in order to provide unbiased tracing of project development. However, very generic metamodels were introduced to designers through project briefs. In project briefs the general ontology of entities with which designers had to deal with was always descriptively

[1] Moggridge, *Designing Interactions*.

introduced. This ontology included the site and its elements, project development process participants; it might have included program components or descriptions of inhabitants. Often the ontology also included an indication as of a strategy for defining project components. In all projects it was implied that a multi-component approach is to be used for developing building forms. During the process of project development, designers would further enhance this generally implied ontology with additional kinds (classes) of developed system components that would suit the specificity of chosen concept and direction of project development.

What was observed was certain reluctance of designers to ontological conventions unless these conventions became an integral part of the design concept, or, unless the convention became the integral part of the employed design environment (instrument). In other cases designers would prefer to consider the projects as one, continuous entity or neglect suggested ontologies as "constraining their creativity" [1].

On the other hand underspecified initial ontologies often resulted in designers developing the project in one direction and neglecting others, or becoming "overwhelmed" or "confused" and not being able to work efficiently. Overcoming reluctance and adherence to the distributed convention always resulted in significantly more robust and complex project outcomes. However, in some such cases, designers explained these phases of project development as "boring" or "unexciting".

The disadvantages of underspecified open metamodels presented themselves mostly in later project stages. Not having a clear project structure significantly slows deployment of a material system. Most significantly, however, the convergence of projects and models developed without shared ontology, or with little ontological overlap proved impossible[2].

o Discussion

The five examples show five different kinds of metamodels that could potentially be employed to develop complex iA systems. IFC provides a highly specific set of classes fully tailored to static architecture. Dependencies between IFC are largely predetermined. As extensive and detailed as they are, IFC only describe the design and construction process and entities (building parts, human actors, management entities etc.) involved in that process. IFC does not support building operation, which renders it incapable to model iA systems. What's more, although the building process can be modelled in the latest release of IFC, it is always being modelled in a predetermined manner. This means that adaptive and emergent qualities could not be inherently contained by architectural systems modelled with IFC. However, a state of an iA building can be potentially modelled using IFC.

UML and SysML allow any kind of objects definitions, distinctively defined in classes or blocks (corresponding to UML classes and components). Blocks can also be nested. Consequently SysML could be used to model any kind of an architectural entity or assembly of components, being building component, user or space, non-embodied component or even a virtual or physical agent environment. In this way UML and SysML provide a significantly more suitable platform than IFC. UML generally allows more flexibility, while SysML is better suited for embedded iA applications, where some of the blocks would be permanently defined, similarly to how classes are defined in IFC.

[1] As indicated in several post-project surveys

[2] E.g. iLounge project

The disadvantage of SysML, however lies in its implementation techniques rather than in its metamodel. The SysML diagrams, as much as UML diagrams, are tailored to modelling of systems with predefined functionality. No diagrams or conventions exist to model system evolution. Once designed (architected) a system is not expected to develop further on its own beyond predefined scenarios (e.g. use cases).

Multi-agent system ontologies naturally appear most applicable to the applied in design research case study examples. However, no established conventions exist in this domain. What's more multi-agent modelling platforms tend to be very function specific and don't allow extensive customizability or design flexibility.

The ad-hoc metamodels are most flexible. They allow focus on most important features and qualities of the system, while leaving others underspecified. This underspecification may be the reason why designers value ad-hoc models as giving most freedom. On the other hand, deciding upon the fundamental ontology of every project individually is counterproductive and does not allow interoperability between models. Ad-hoc modelling allows fast start, but slows down more complex project development.

In the last group of metamodels an open specification of classes proved useful, but not sufficient. It can be concluded that a generic kind of a metamodel is essential, however detailed specification of system ontology may be too constraining for system development. An alternative needs to be found in which projects can freely develop and evolve, while maintaining consistency in their fundamental structure, allowing for divergence and convergence of developed models and systems.

4. Conclusions

The chapter has set off with the premise to investigate how to deal with interactive architecture considered as a system made up of large numbers of interacting entities and to draw conclusions for the impact these consideration have on the foundations for the iA development framework.

4.1. Summary

- Architecture can be considered to be an open system of interacting entities.
- Architectural systems contain humans, artificial and natural actors.
- Numerous phenomena, such as emergence or feedback loops broadly studied in complexity science apply to architectural systems, and gain additional relevance when applied to interactive architecture.
- Any kind of actors in architectural systems can exhibit autonomous agency. Agents in interactive architectural systems are expected to exhibit higher degrees of agency.
- Development and evolution are inherent in complex adaptive architectural systems.
- Design and operation of complex adaptive interactive architecture are intertwined in one seamless process.

4.2. Expectations

Approaching interactive architecture as a system of a very large number of interactive components breaks down the established perception of buildings as highly integrated, singular entities. Instead, a building is seen as an ecosystem of building components, appliances, installations, furniture, but also including its inhabitants, stakeholders or experts involved in its creation.

This view opens a multitude of novel opportunities and allows for independent study, formation and development of interactions between all actors in such complex building systems. As a consequence, buildings can be seen as ecosystems, expected to be capable of development, evolution, growth and adaptation.

4.3. Risks

The proposed changes in the understanding of building systems potentially allows for adaptation of buildings through very large numbers of specialised interactions. This approach, however, contains significant risks:

- Complex systems have the tendency to fall into higher entropy states through positive feedback loops. Balanced systems need to be developed though negative feedback loops.
- Conscious creation of architecture as a complex adaptive system means giving up full control over such system.
- Complex systems with heterogeneous agents are difficult to comprehend for humans, thus also to design and use from outside of the system's internal logic.
- No cultural models exist for dealing with complex adaptive interactive architectural systems. It is unknown if such systems will be culturally accepted and how to communicate the dynamic affordances of such systems to their users.

4.4. Challenges and Opportunities

The challenge that results from approaching architecture as complex adaptive interactive system is to transform the risks related to giving up control and emergent properties of distributed systems into their qualities. The unpredictability of complex adaptive systems allows for these systems to adapt to circumstances unforeseen during the moment of deployment of these systems. The qualities that emerge from such unpredictable adaptations are manifold. They can cater to features such as spatial experience involving surprise and unpredictability or to functional aspects such as adjustment of spatial affordances to current activities of users or crowd direction in planned or emergency scenarios. They can also be employed to proactively influence inhabitant activities or needs, in which area ethical concerns need to be considered.

The role of designers and expert in these systems is henceforth expected to change. Instead of delivering finished system solutions, they will steer, maintain and expand continuously running and changing architectural systems.

The biggest opportunity provided by complex adaptive interactive architecture is its inherent reliability and resulting robustness in dealing with external and internal unpredicted changes that also include errors or failures. In traditional centralised and hierarchical systems failure of one element in the linear chain of command renders the entire chain dysfunctional. In a distributed adaptive network of elements, working elements can take over the functions of those that failed.

4.5. Problems

The lack of shared ontology for complex adaptive interactive architecture presents itself as the main bottleneck in its further development of iA. Traditionally established architectural notions and roles devaluate and change meaning when the worldview on architecture based on largely distributed complex systems is introduced.

What follows is the lack of relevant methods, techniques and instruments that would inherently work in that worldview. As a result, language of traditional, permanence-oriented architecture intertwines with the language of interactive architecture, leading to misunderstandings among designers, experts and users.

Despite significant recent interest in interactive architecture, it is common for individual designers to focus on isolated aspects of iA such as smart materials, embedded computing, sensor data, kinetic actuation or interaction and user experience design, or conceptual future visions. As a result, realised interactive installations deliver fragmented perception about interactive architecture, typically referred back to traditional worldview on architecture and existing building conventions and typologies. Conversely, the actual benefit of autonomous architectural adaptation being an inhabitant driven, continuous and distributed process becomes lost. To date no comprehensive example of an iA system has not been developed. Consequently, such lack of holistic approach towards development of iA significantly hinders its further development.

IV. Tracing design research case studies

Summary

The chapter starts with a revision of the initial research framework. The chosen research approach follows the adapted grounded theory research method for building up of the iA development framework. Experimental, exploratory design case study research is employed as source of qualitative and, to a lesser extent, quantitative research data and provides means for systematic validation and adjustment of the iA framework throughout the process of its gradual formation through this and following chapters.

Consequently, the chapter primarily presents an account of a series of design research case study experiments. The chapter is driven by the initial assumption of a general and purposefully underspecified set of guidelines for creation of complex, adaptive, interactive architecture systems. These guidelines follow research presented in chapters II and III and are founded in critical evaluation of the two state-of-the-art reference projects analysed at the outset of the chapter.

Consequently, throughout the selective tracings of networks of actors building up case study projects, various aspects of a process of developing interactive architecture are iteratively approached, tested and evaluated while selective focus is in turn given to key aspects of these processes. Numerous challenges and strategies, techniques and methods for addressing them are explored. Upon the termination of each project they are either rejected and replaced with new solutions or further improved and refined. This process permits an iterative build-up of a structure for a practical and efficient methodology for interactive architecture.

The chapter starts with an introduction to the method of tracing design processes and rationale behinds its choice (section 1.). Two reference state-of-the-art design cases are accordingly traced to provide a starting point and indicate biggest challenges for interactive architecture design methods (section 2.). Design case studies are subsequently discussed in five categories, corresponding to five aspects of the iA design process; experiential prototyping and realisation of designed systems (section 3.), assembling projects out of autonomous building components (section 4.), involvement of human agents in iA systems (section 5.), design of spatial organisation of complex multi-component systems (section 6.), and largely distributed projects (section 7.). In conclusion, challenges coming from these case studies are discussed, deliberating the role of a designer as working from within the iA system and showing the critical role of design instruments and experiential prototyping, which are to be further investigated in chapters V and VI.

1. Development of an iA framework through case study research

The chapter sets off to investigate the question how design processes for creation of interactive architecture should be structured? It has been established, that such processes need to be different from what can be encountered in traditional architectural praxis. Following Biloria's argument "The convention of working top-down with form and styling as leading parameters as opposed to understanding the integrated process of emergence from a bottom-up perspective where raw data, processed information and their relation to material systems are used to generate contemporary, sustainable and performance driven architectural formations will certainly pave the future direction of design economies."[1] However, initially little can be assumed about what exactly these processes should entail. Specific strategies, design practices and eventually design methods need to be iteratively tested and evaluated in order to build up design knowledge in this novel design territory.

Upon investigating the state-of-the-art in architecture-related technology, namely building management systems, sensor networks, media facades, building automation systems or smart materials, it is clear that most technological "ingredients" required for interactive architecture to be realised already exist. There is also a significant demand for spatial adaptability and transformation of existing buildings, followed by numerous precedent examples and attempts of creation of adaptable architecture[2]. The new paradigm of systems science that has found its way to most scientific disciplines and much can be learned from it in respect to the ways in which complex problems encountered by interactive architectural systems could be dealt with[3]. New, digitally driven, possibilities for virtual and physical creation of interactive architecture are already being broadly explored and many more lessons from systems engineering, computer science and interaction design can still be learned.

The radically new spatial qualities that dynamic buildings have potential to deliver require a new approach, unconstrained by past conventions and standards. Yet, there is still a notable lack of integration of all the above mentioned developments and no sound design methods nor frameworks exist that could further facilitate the development of interactive architecture. New, appropriate methodologies need to be defined and validated. Implications of creation and use of complex interactive architecture require thorough investigation and experimentation before applicable and reliable solutions can be brought to real-world applications.

1.1. Research problems and opportunities

In the second chapter it has been argued that in order to create architectural habitats that flourish instead of gradually declining, these habitats need to be able to locally adapt to changing external and internal conditions, driven by changing needs of their human participants. This process can be observed on the urban scale in historical as well as existing cities, yet may not be fast or granular enough to cope with the speed and magnitude of changes in use patterns in contemporary habitats.

[1] Nimish Biloria, 'InfoMatters, a Multi-agent Systems Approach for Generating Performative Architectural Formations', *International Journal of Architectural Computing* 9, no. 3 (September 1, 2011): 205–222.

[2] For references see section II.2

[3] For references see sections III.1 and III.2.5

In the third chapter, the concept of approaching architecture as a complex system has been discussed, leading to the conclusion that architectural systems should include not only material, non-living objects, but also humans, other living beings and immaterial entities. Consequently, as argued, interactive architecture should be treated as a complex adaptive system, where all its components interact, transform and adapt to each other leading to emergence of a higher order adaptation of such system as a whole. This approach required a new look at architectural design methods and has led to a postulate for a hybridized approach, where distinctions between designing and operation, as well as between designers and users became dissolved.

Autonomous adaptation of architectural environments to unanticipated internal and external conditions can be achieved through development of those environments as complex adaptive multi-agent systems. In highly intricate conditions, which are typical to architectural design problems, such approach towards architectural adaptation is expected to be more efficient than currently practiced human-controlled or automated adaptation of architectural qualities, what should be proven both theoretically and experimentally.

In consequence, the initial hypothesis can be stated that hybridized development of architecture as complex adaptive systems build of a high number of autonomous, adaptive agents under well-engineered conditions will allow development of highly adaptive architectural processes perpetuating the flourishing qualities of human habitats. Treating interactive architecture as a multi-agent system (MAS) can deliver appropriate foundations for an integrated yet non-constraining methodology for creative development of iA. Concurrent spatial design, interaction design, development, engineering and operation of complex adaptive architectural environments can be integrated in one multi-agent system. Application of such system should allow substantial increase of widely understood performance (cultural, functional, structural and energy consumption) of adaptive interactive architectural environments.

Development of full-scale interactive architectural interventions operating as complex adaptive systems requires a new framework permitting integrated and concurrent design, production and operation of such architecture. Development and empirical validation of a comprehensive set of exemplary methods, tools, techniques and models can serve as a proof and an open foundation for such framework and initiation of further research into this domain.

The goal set for research presented in the following chapters is to develop and validate an integrated framework for creation of complex, adaptive and interactive architecture, approached as a multi-agent system.

Main principles of operation of a proposed system are thus inherently simple. Difficulty lies in finding specific solutions for virtually designing, validating and deploying those systems in physical realm. The chosen way to validate proposed system logic is to conduct a set of experiments to apply various concepts of system architecture to actual spatial scenarios and validate its operation in a semi-controlled environment.

1.2. Framework development methodology

In the context of the presented argument, too little knowledge exists to postulate any specific framework for integrated design and operation of iA, which could then be proven or disproved as successful through a set of devised design experiments, what would have been the most common way of conducting scientific research and applying the scientific method[1].

[1] Creswell, *Research Design*.

In social sciences, and alternative approach has been postulated under the name of the grounded theory. In grounded theory research, no theory is assumed as hypothesis at the outset of research. Instead, as research experiments are being conducted "codes" are being extracted as patterns of collected data, consequently concepts are formulated, categories are made and eventually the theory is formulated *a posteriori* to the conducted research experiments.[1]

In case of grounded theory, the aim of theories developed in the research process is to develop an understanding of a phenomenon (often of social nature). In case of research on complex adaptive architectural systems, the analysed phenomenon, which is the creation and operation of complex adaptive systems, depends on the formulated theory. This adds a complication to the research methodology, but it does not contradict it. However, it requires an iterative approach when working versions of the theory are postulated along the research experiments and the application of these working versions is validated as the theory gains its shape.

In line with the general consideration for the grounded theory research method, actor-network theory provides the ontological foundation for the construction and navigation through the design research experiments. Actor network-theory has in itself been shown to be a valid design research methodology[2]. It provides thorough tools to analyse complex social phenomena and trace networks of dependencies, interactions and transformations through studied situations, without reducing them to constrained systems a priori to the conducted research. Tracing actor-networks can thus become a tool in which phenomena observed in design research can be mapped and consequently system "views" can be derived from such tracings as intentional reductions and generalisations of what was being traced.

For the definition of research experiments, design case study research has been chosen. As discussed by Richard Foqué, "Research by design tries to explore and change the world, and by doing so, tries to gain knowledge about how man analyses and explores the world and brings it into culture: how we create a man-made world. It does so by creating design applications, relying on technological knowledge and artistic interpretation"[3] Design research case studies are to be formulated in ways, that through their execution, a new insight is granted into the studied knowledge domain and new models can be constructed, contributing to the advancement of theory. Experimental, exploratory design case study research can thus be employed as source of qualitative and, to a lesser extent, quantitative research data and provides means for systematic validation of developed theory throughout the process of its formulation

1.3. Framework development plan

In the context of presented problems, the framework for interactive architectural systems takes the role of the theory that is to be iteratively developed through a series of design case study experiments. The domain of research is the integrated design and creation of operational architectural systems. The operational architectural systems are set to be formed and studied as systems consisting of heterogeneous adaptive agents. These agents can be building components, humans and other living entities as well as non-embodied entities.

[1] Charmaz, *Constructing Grounded Theory*.

[2] Kraal, 'Actor-network Inspired Design Research'.

[3] Foque, *Building Knowledge in Architecture*.

In consideration to this approach, the architectural system is acknowledged to be one of many possible views on constructed reality of architecture. The framework in which created systems operate is the subject of exploration and therefore it is open and extensible. The very nature of that framework is unknown and will be defined throughout the experiments

Following the listed assumptions research experiments will be conducted in three focus categories, namely design methods, instruments and operation, the three categories will be cross-influenced. Throughout these experiments a hybridized framework for creation of interactive architecture will be developed and subsequently discussed in detail. The final research case study experiment will be an attempt to apply and evaluate the framework and will serve as foundation for critical discussion and an outlook into future.

1.4. Setting up design research case studies

In order to provide the initial reference point for the following design case-study driven research, two projects have been initially selected as state-of-the-art design process references positioned on the path towards interactive architecture.

The first of these projects is Cockpit and Sound Barrier, the signature work of the architecture office Oosterhuis_Lénárd[ONL]. Although it is not an interactive building, its original design included interactive features. The project demonstrates the new trend in architectural design, involving parametric modelling and mass-customization of project components. Its design process illustrates a way of dealing with an architectural project as a system of unique components, and the collaborative process of its development, joining designers and engineers in one horizontally structured team.

The second project is Trans-ports, the project by Oosterhuis_Lénárd[ONL] later joined by the Hyperbody group. The Muscle Trans-ports interactive installation being the last iteration of the project has been a central piece of the “Architectures non-standards” exhibition at Centre Georges Pompidou in Paris, 2003. The exhibition has become an emblematic event for the contemporary architecture, bringing together forward-thinking practitioners reinventing the architectural praxis to match present day challenges and opportunities. As a result, Muscle Trans-ports has become one of the international icons of interactive architecture for years to come.

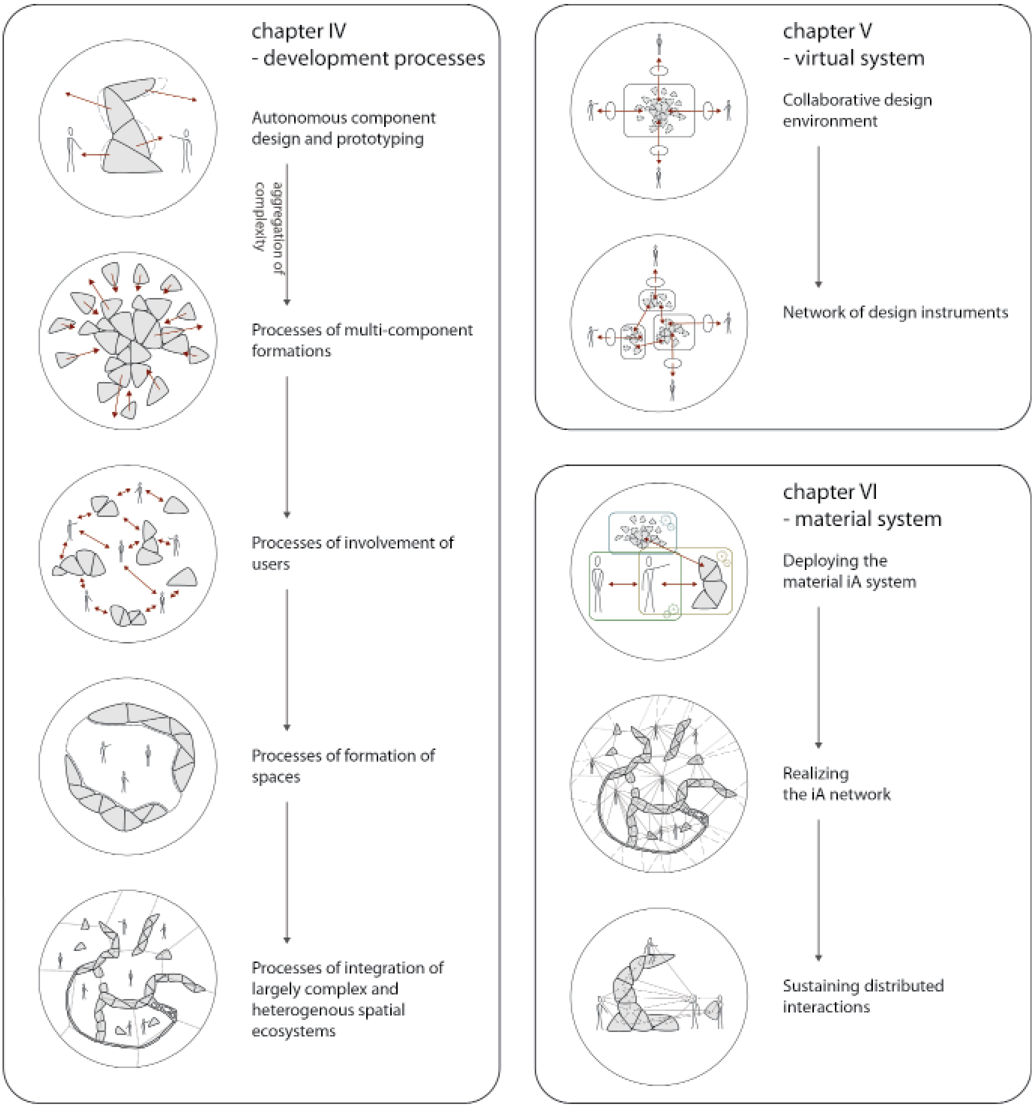

chapter IV
- development processes
Autonomous component design and prototyping
aggregation of complexity
Processes of multi-component formations
Processes of involvement of users
Processes of formation of spaces
Processes of integration of largely complex and heterogenous spatial ecosystems
chapter V
- virtual system
Collaborative design environment
Network of design instruments
chapter VI
- material system
Deploying the material iA system
Realizing the iA network
Sustaining distributed interactions

Fig.19. Organization of themes for tracing design research case study projects in chapters IV-VI.

All projects in this chapter are studied using the tracing method, inspired by Bruno Latour's Actor-Network Theory[1]. The complexity of these projects is too high to identify all system ingredients and all relations between them. The boundary of the projects is equally hard to define, as designed system highly depend on involvement of kinds of actors (human and non-human) that traditionally are not considered to be inherently parts of architecture. For these reasons, tracing the "actor-network" of projects can be used as an instrument for better understanding of the processes of these projects' formation and eventual definition of these projects as finite systems. Each project description is separated into three or more sub-points. The first "Challenge" sub-point traces the events precedent to the project, leading towards the moment of project initiation. Consequently the process of the development of

[1] Latour, *Reassembling the Social*.

the project is traced in one or several following sub-point. The last "Result" sub-point discusses the outcome of the project at its assumed end-moment and traces its further development, followed by concluding the description and summarizing the findings.

Each section ends with a discussion, summarizing all the findings in a list of bulleted points. Throughout all chapters a simple graphical convention is used to represent and bind these conclusions across the chapters.

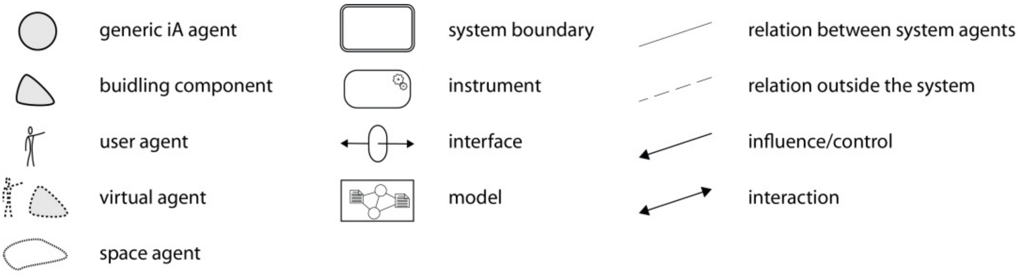


Fig.20. The graphical language of identified design process elements.

Following the critical evaluation of the development of Cockpit-Sound Barrier and Muscle Trans-ports, a series of design research case studies is initiated. The purpose of these projects is to iteratively augment and transform the design processes employed in reference projects, in search for appropriate design methodology for complex adaptive interactive architecture. Case studies take place in two contexts. One is the academic context of the Hyperbody MSc programme at the Delft University of Technology, where a broader range of projects can be explored in educational design studio setting. The other is the context of the commercial practice of Oosterhuis_Lénárd[ONL], where projects can be investigated in direct relation to real-world scenarios. All studied projects have been traced in order to devise key aspects and variations of employed in them design strategies and to gain insights into complex interaction patterns, leading to formation of stable and robust architectural systems.

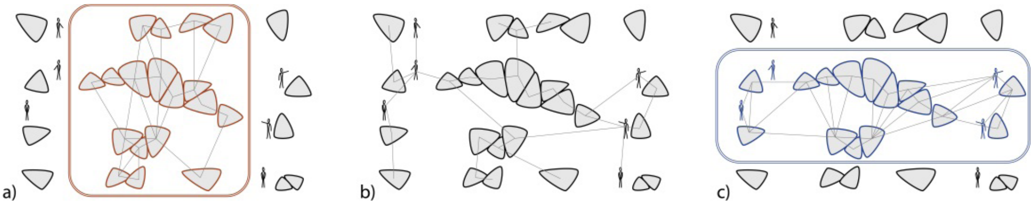


Fig.21. a) Initially established open system (elements, relations, boundary, interactions with the outside) b) Tracing of actors c) Resulting new open system.

In each of the described projects all project components and participants were treated as autonomous agents (actors). The development of the projects has been traced without consideration for any layers of abstraction. In the first group of discussed reference projects, the focus is given to assemblages of material building components. Subsequently, case study research projects illustrate involvement of humans, spaces and non-embodied entities as agents in developed systems. Eventually, assemblages of all such types are looked upon. Ultimately, processes of physical deployment of iA systems are analysed.

2. Reference projects

Challenges:

- State-of-the art methods for designing highly complex and adaptive complex adaptive require investigation.
- Direction for advancement of those methods required to increase the complexity and adaptivity of designed architecture need to be identified.

As discussed in chapter III -section 1.2, traditionally buildings are assembled out of basic components (prefabricated and modular or made on-site such as bricks, stones, wooden beams, metal struts, concrete slabs and many others). Such assembling follows plans earlier prepared by architects and validated by building engineers. It is carried out by contractors in ways historically established through shared building conventions.

Since late XXth century, an alternative approach has developed and is contemporarily radically changing such traditionally established design and construction processes. Through the employment of computer design tools, "parametric designing" has emerged. In combination with computer numerically controlled (CNC) production it has led to the new trend of "mass customisation" of building parts and resulting emergence of "non-standard architecture"; creation of buildings where every building component is individually described and defined in relation to other components, engineered and produced with high precision to the exact and unique measurement. Design strategies such as ONL's "one building-one detail" naturally follow and the Cockpit and Sound Barrier project is the flagship example.

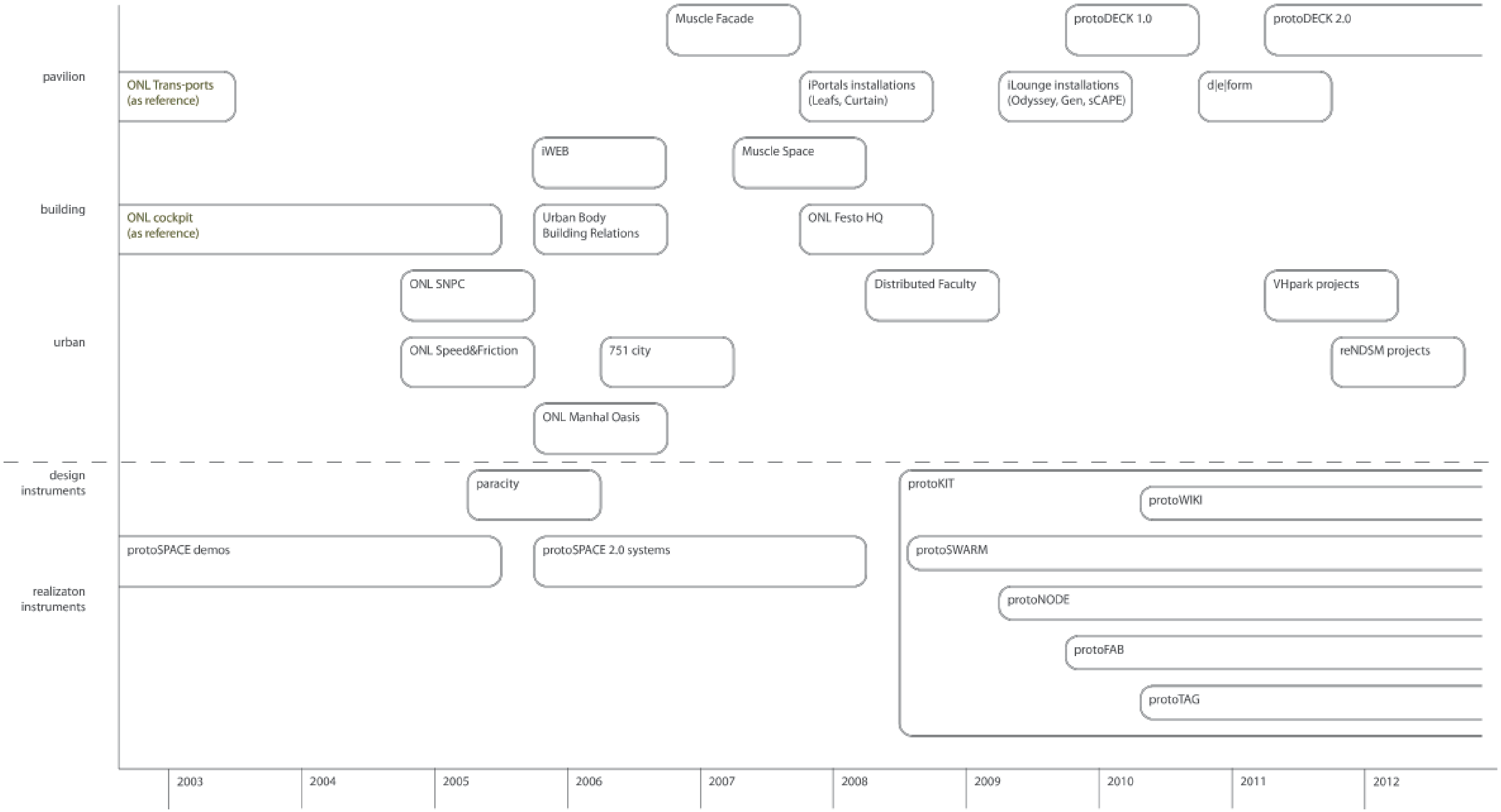
Another contemporary trend tangential to "mass customization" could be broadly referred to as "building activation", encompassing augmentation of buildings with sensing, information processing and actuating capabilities. This trend can be already observed in a variety of realised projects that range from "automated buildings" to "media facades". However, most contemporary developments point at more integrated applications, where sensing, processing and actuation are to be inherently included in buildings rather than added as a post-design layer. What follows, is adistributed approach to such system's operation where no central control is enforced on building's behaviour. The Muscle Trans-Ports installation is an internationally recognised example of this trend.

As discussed in chapter III, it is clear that further development of interactive architecture stems from the intersection of the two trends of "mass-customization" and "building activation". The tracing of the development of two reference projects being state-of-the-art examples in these two domains is consequently expected to provide and insight for better understanding of design processes of both kinds of architecture and to lay grounds for further formulation of design strategies for interactive architecture emerging out of integration of the two trends.

2.1. The Cockpit and Sound Barrier

Oosterhuis_Lénárd [ONL] is an architecture firm, which has positioned itself in the avant-garde of the development of non-standard architecture. ONL's signature projects: The Cockpit and Sound Barrier can be provided as a clear illustration of these new trends. These projects can also be used to demonstrate how non-standard architectural projects can be traced as a network of people, and virtual and material building components, and how such networks can consequently be viewed as complex adaptive systems, leading to consequent discussion on ways in which building adaptation can be performed within such systems.

The author has not participated himself in the design or building of the Cockpit and Sound Barrier projects, but has collected accounts of these processes from many of its direct participants. The following description traces interactions in the design and materialisation



Img. 4. Design research case study projects - chronological overview

processes of these buildings, which were unfolding among designers, clients, building components and architectural spaces. These accounts and related discussion shall serve as a point of reference and departure for following it design research case studies.

a) Outset

The Cockpit and Sound Barrier projects were initiated through the interaction between three parties. The ONL office pro-actively proposed the project idea to the municipality of Utrecht Leidsche Rijn and the Hessing BV. The challenges that ONL proposed an answer for were questions of how to create a sound barrier separating the Leidsche Rijn area from the busy A2 highway while providing an attractive landscape and a symbol for the area behind the sound barrier without reducing the commercially important visibility from the highway of the Hessing BV car showroom planned in the area? Briefs for the two projects were formulated in mutual dependence. The Sound Barrier was eventually commissioned by Utrecht Leidsche Rijn to acoustically isolate the Leidsche Rijn neighbourhood from the A2 highway on the stretch of 1,5km and to do provide a unique architectural landscape object. The Cockpit was commissioned by Heesing BV to become a building fit in between the two segments of the Sound Barrier and containing a car showroom exposed to the cars passing along the highway.

b) Process

The development of both projects progressed in parallel and in tight relation to each other. Prof. Kas Oosterhuis took the role of the lead designer and together with his partner, visual artist Ilona Lénárd they have virtually drawn three three-dimensional Bezier curves, referred to as "powerlines"[1], onto the site of the otherwise at that time "uninformed" projects. These powerlines could be seen as virtual agents created with the purpose to organise the designed form and functions and operate as attractors attributing intentional styling to the project. The powerlines followed the line of the neighbouring highway, but they also introduced their own, unique dynamics.

[1] Ilona Lénárd, 'Powerlines', in *iA#2* (Episode Publishers, Rotterdam, 2009).

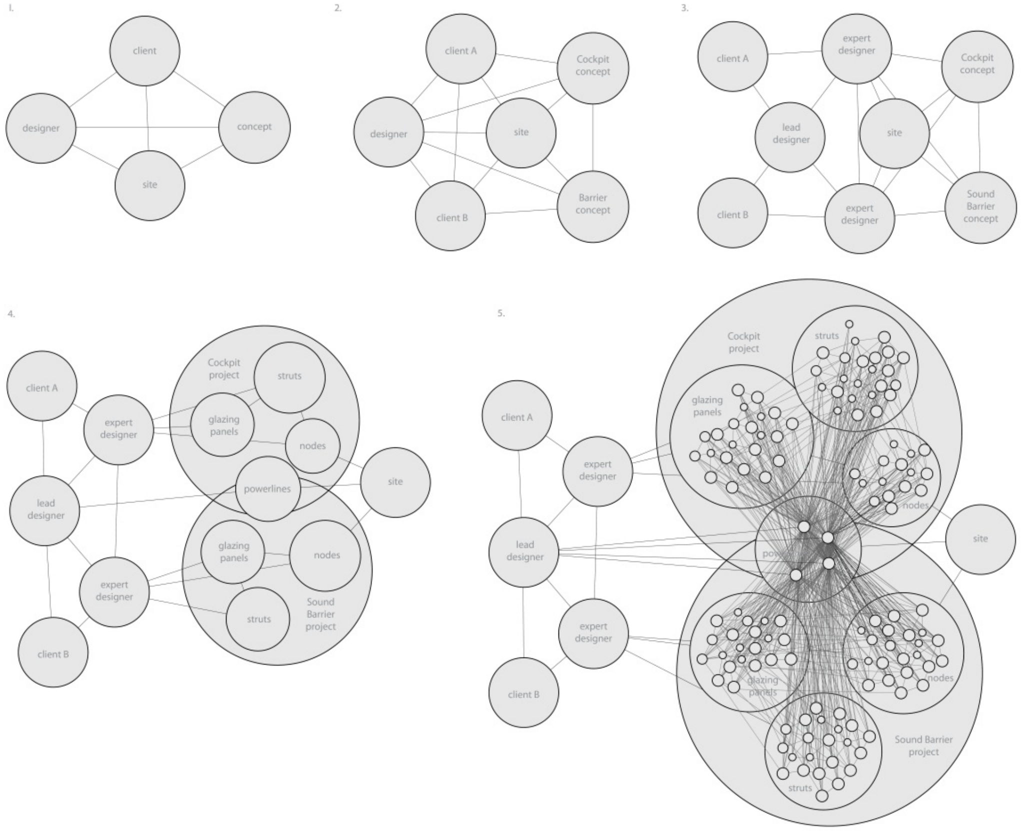
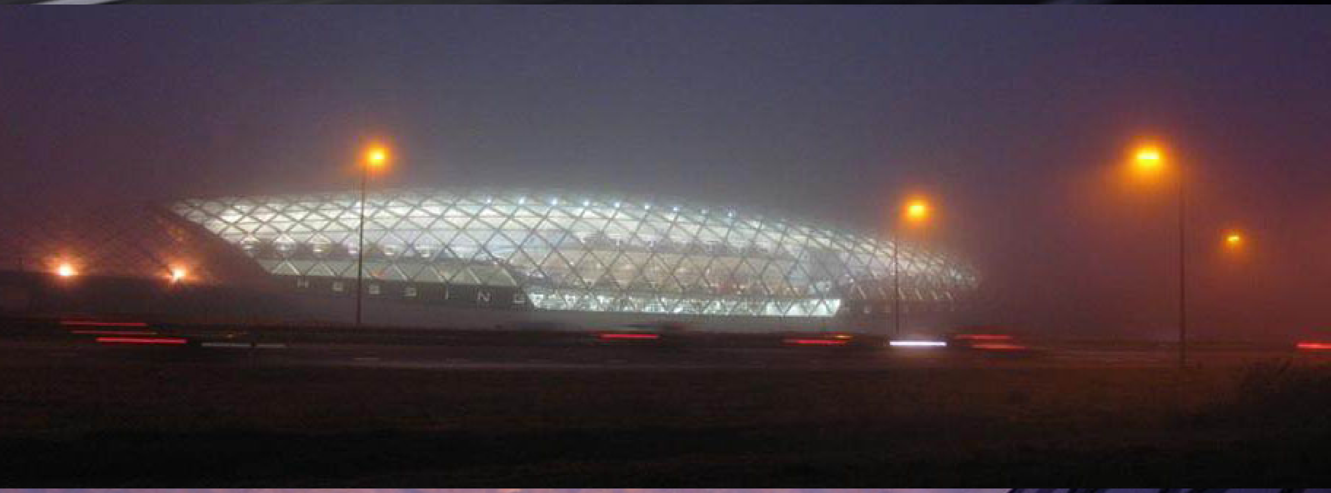
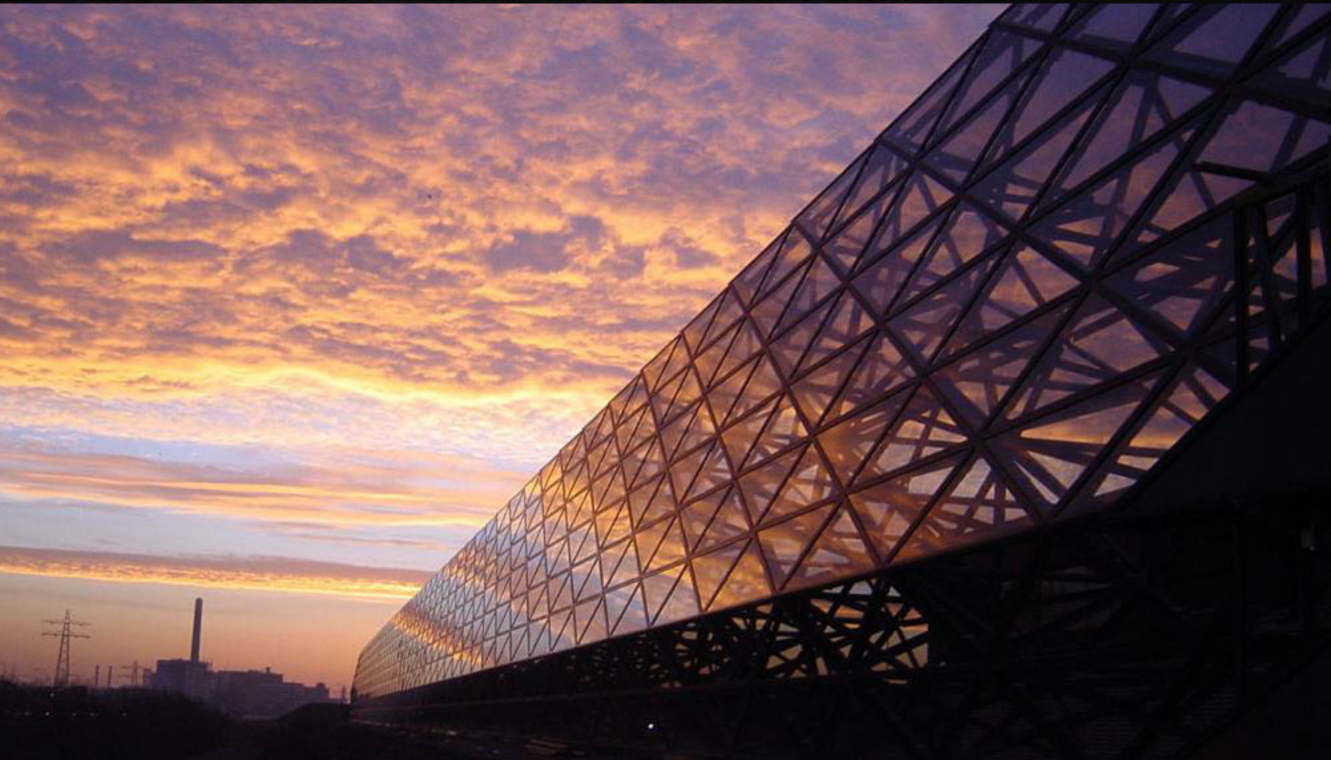


Fig.22. Interpretation of initial steps of development of the Cockpit and Sound Barrier treated as a system of heterogeneous agents developing over time.

Consequently, the team of ONL architects and engineers with the leading roles of Sander Boer, Gijs Joosen and Cas Aalbers, created approximately 7000 and 2000 point objects in the Sound Barrier and Cockpit projects respectively, filling the virtual space already occupied by the powerlines. These points were subsequently provided which logics to form an even distribution and “populate” surfaces following the initially drawn powerlines. Steel struts, nodes and glass panels were then generated to fill the spaces between points in both projects, based on distinct sets of rules and properties within each of the two projects. A looping floor slab was then designed and fit parametrically into the Cockpit, organising the continuous showroom space. Following this process adjustments were made iteratively in conjunction with numerous discussions among clients and architects. Parametric logic embedded in the created system allowed to implement these adjustments quickly and flexibly. The detailing of the project followed the parametric logics. Eventually the production data for steel components and glass panels was extracted directly from the system and submitted to Meijers Staalbouw and Pilkington factories, where the 40000 unique elements were produced to the exact measure, uniquely labelled and with resulting increased ease assembled on site by the contractor teams.

The Cockpit and Sound barrier design has developed to include a dynamic lighting solution, where every node of the front side of the structure was envisioned to include a multi-colour light source. Through the sensing of traffic and communication with adjacent nodes an emergent light pattern has been envisioned that would create a dynamic relationship between the buildings and the ever changing A2 highway. However, problems were encountered in further development of this vision as a) there were no technical possibilities available to the





team to design and simulate the behaviour of thousands of such nodes b) the engineering cost of the technical solution has surpassed the project budget. Consequently in the shared decision between clients and designers the vision has been withdrawn from the concept.

c) Result

The construction of Cockpit and Sound Barrier has been completed in 2005. The car showroom has performed its role with success, becoming an icon for high-end car retail, commonly recognised in the Netherlands. Although the original client Hessing BV has gone bankrupt in 2011, the Cockpit has retained its function under the new brand “Louwman” without any changes to the building or its organisation.

The process in which the Cockpit and Sound Barrier were formed is a seminal example of non-standard architecture. Activities of designing, engineering, producing and assembling of inter-dependent building systems are densely interwoven with each other in one integrated process. The building system develops throughout the entire process and involves intensive, complex interactions with architects and engineers, but also with clients and with elements of the site, such as the highway system including driving on its cars, or the aggregations of people, offices and houses forming the Leidsche Rijn area. Different parties interact with different parts of the project on different scales and in different ways. They also interact among each other. Building parts are connected through digitally scripted relations. Once produced and assembled, these virtual relations become transformed into physical interactions.

The tracing of the two projects over time reveals the continuous growth and development of the designed system and differentiation and proliferation of its components (figure 1.). Many analogies can be found between this process and a growth of a living organism, where cells specialise while the organism grows. Projects begin as a system of only four interacting actors. Initial design sketches rapidly develop into a complex assembly of virtual and abstract geometric entities (such as powerlines and points), these entities are gradually given more virtual body, details, parameters, and are eventually fabricated and assembled into a physical building. The transformations between process phases are not creating discontinuities in the architectural system, but bifurcations, occurring when the system transforms from one stable state to another and its new qualities develop in an emergent, but intentionally guided manner.

Comparing to a traditional design and construction process, the process can be characterised by:

- No complexity “jumps”. The complexity of the project increases gradually, starting in early design phase and is equally present in all aspects of the project. The high complexity of the projects is not reduced in or after the fabrication process but remains the feature of the project
- Largely increased number of design-engineering iterations involving deigned system optimization and adjustment until the moment of fabrication and assembly
- Largely shortened time of a single design-engineering cycle
- Largely shortened time of project construction and assembly
- Largely reduced number of assembly errors
- Close involvement of clients in design and instant inclusion of client feedback in design iterations

Img. 5. Left, from top: ONL[Oosterhuis_ Lénárd] Cockpit design including interactive lighting, Cockpit realisation, Sound Barrier realisation

2.2. Muscle Trans-Ports

Trans-ports is a project by ONL[Oosterhuis_Lénárd] that has had multiple reincarnations. Muscle Trans-ports, more commonly referred to as Muscle NSA[1] is the last version of the Trans-ports project, which took the form of the exhibition installation.

a) Outset

The Trans-ports project has been initially conceived and revealed in 1999. The original idea emerged during discussions between Kas Oosterhuis, Ilona Lénárd and Marcos Novak. Trans-ports idea was first publicly presented by Kas Oosterhuis at the first Archilab conference as "a building that would change shape and content in real time", while being part of "a swarm of buildings, distributed around the world, communicating with each other"[2]. As then envisioned, trans-ports buildings would be equally present in the physical and virtual reality. They would permit its visitors to immerse themselves in the virtual world, interact with both physical and virtual space surrounding them and virtually connect to other remote trans-ports buildings and their users. The idea did not explicitly focus on a particular function of such building. Its aim has rather been to present a different, new way in which architecture can be made and new kinds of experiences and interactions such architecture can provide.

The initial concept did not explicitly show a solution to how such buildings could be constructed. However, presented renderings indicated employment of kinetically active and highly stretchable surface material. Such material could be hypothetically created using electrocative polymers, which are contemporarily still not commercially available, yet ample experiments point at its commercial availability in the coming years. A more future-oriented option would be employment of "programmable matter"[3], which technology has been hypothetically proposed as attainable within next decades[4] and has been intensively researched upon since 1990s. However, to date it does not exist and is unlikely to be available within the coming decade or more. The main challenge faced by the trans-ports project has thus been on finding an existing technology for being able to realise this highly innovative concept.

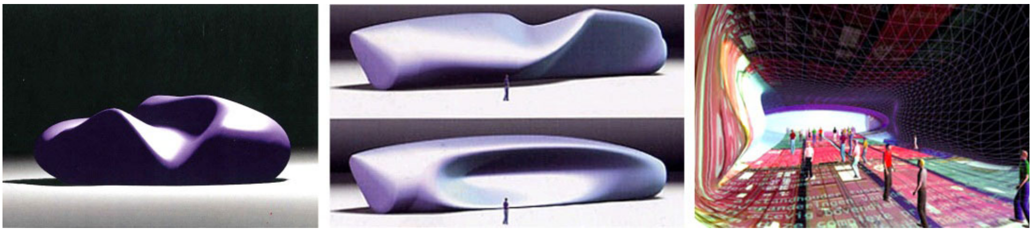


Fig.23. The first version of the Trans-ports concept, 1999 © ONL[Oosterhuis_ Lénárd]

b) Process

The Archilab conference presentation can be seen as the initiation of the trans-ports' "virtual" existence. The concept itself, although originated and shaped by Kas Oosterhuis, has been highly influenced by discussions with numerous actors, starting with Ilona Lénárd and Marcos Novak, but extending to many other professionals and visionaries grouped around Archilab and ONL. Another influential actor for the development of Trans-ports were the conference-

[1] Chris Kievid and Kas Oosterhuis, 'Muscle NSA, a Basis for a True Paradigm Shift in Architecture', in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012).

[2] Kas Oosterhuis, *ONLogic: Speed and Vision The Master Architect Series* (Images Publishing Group Pty. Ltd., 2008).

[3] Goldstein, Campbell, and Mowry, 'Programmable Matter'.

[4] Nicholas Fisk, *Trillions*, New edition (Hodder Children's Books, 1998).

exhibition events which provided stimuli to showcase the idea and to spur discussions driving its further development. After Archilab, the next such event was the Venice Biennale exhibition, where the next version of Trans-ports was showcased.

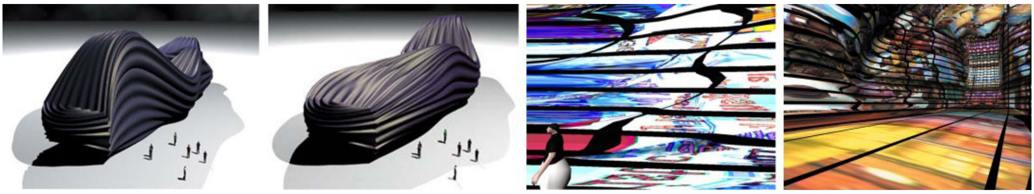


Fig.24. The second version of the Trans-ports concept, 2000 © ONL[Oosterhuis_ Lénárd]

One aspect of the new version of Trans-ports had been the introduction of the harmonica-like undulation of the form, reducing the required stretchability of the project's skin. Although the actuation and structural support mechanism has not been addressed, existing materials, such as thick rubber would have allowed creation of the building.

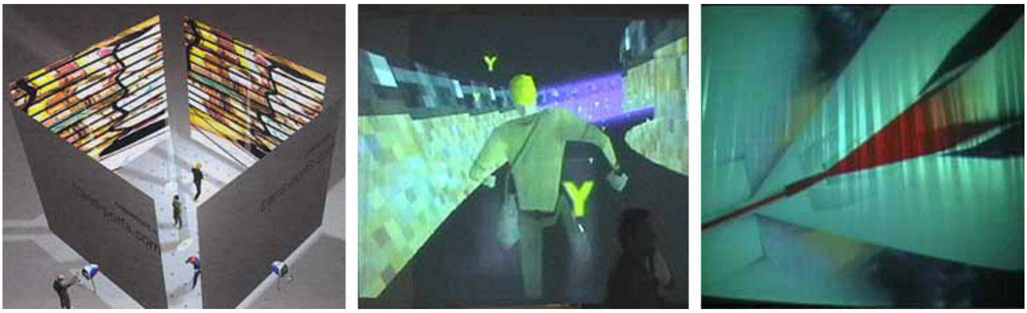


Fig.25. Trans-ports installation at Venice biennale, 2000 © ONL[Oosterhuis_ Lénárd]

The other aspect of the second design iteration has been the behaviour of the building. The trans-ports installation at Venice biennale consisted of a four screen "cave" setup, where the virtual simulation of possible building behaviour was simulated. The simulation included the use of sensor inside the cave providing actual interaction between visitors and the virtual trans-ports interior.

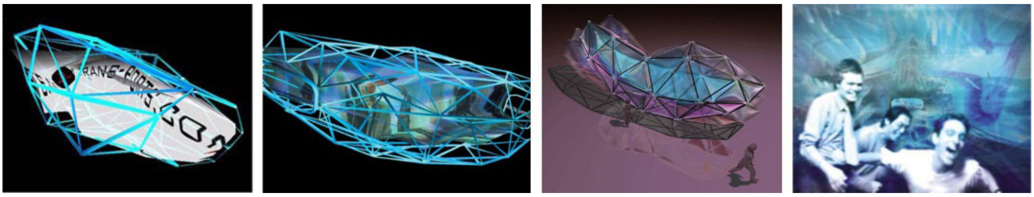


Fig.26. Development of Trans-ports Muscle concept, 2002 © ONL[Oosterhuis_ Lénárd], Hyperbody

The next version of the trans-ports pavilion included several sub-iterations over the design. This step involved students and researchers from then newly formed Hyperbody group. The main focus of that iteration was on the realisation of the project. Initial designs included a kinetic truss structure from which a flexible "cocoon" of trans-ports was suspended. Eventual solution was inspired by research of Hugo Mulder, and a different formal expression and material were chosen. Inflation of the form and pneumatic actuators (Festo fluidic muscles) were used to achieve structural stability and kinetic actuation.

The development of the project was driven by the invitation of ONL to the Architectures Non-Standards exhibition held in 2003 at the Centre Georges Pompidou in Paris complemented by the provided budget and eventual material sponsoring by Festo bv an Buitink bv, which together provided the constraints for the size and scope of the installation to be realised.

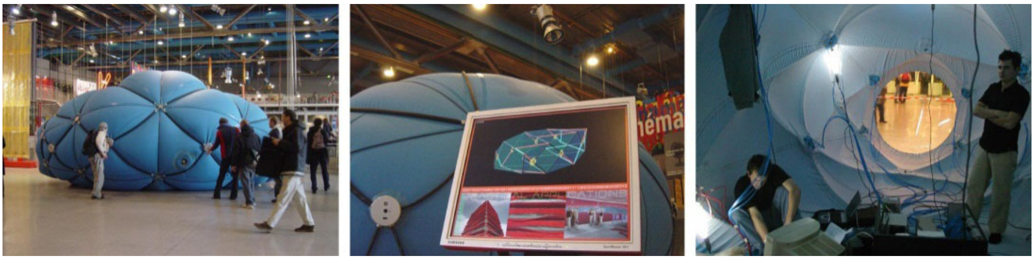


Fig.27. Muscle NSA installation, 2003 © ONL[Oosterhuis_ Lénárd], Hyperbody

The Trans-ports Muscle installation can be seen as result of the evolution of the trans-ports concept. However, other influences can also be noticed. Clearly, the inflated body of the installation has been inspired by the earlier project of ONL, the ParaSITE, realised in 1996. The inflated shape has been wrapped by a mesh of fluidic muscle actuators, while eight of the connecting nodes were equipped with proximity and touch sensors. The interior, unlike in the general trans-ports vision, was not accessible for the public and hosted the control unit of the installation. The control unit hosted the virtual representation of the installation programmed in Virtools DEV software. The representation included all actuators and sensors operating as autonomous agents, in real time receiving data from the actual sensors, exchanging data with each other, and triggering contraction or relaxation of fluidic muscles.

The collective behaviour of the virtual agents linked to the behaviour of the actual installation has led to a life-like and unpredictable behaviour of the entire structure. "The result of the various interacting layers of the programming is a behaviour that is sensitive and slightly unpredictable, only controllable to a certain degree, according to the principles of a multi-valued fuzzy logic"[1]. Visitors to the exhibition could learn over time how some of their activities were affecting the fluid behaviour of the structure. As evaluated by visitors and critics, the installation has shown an entirely unprecedented dynamic space. However because of the inability to enter, it has been rarely perceived as a building or architectural space. Former student of Hyperbody Chris Kievid, working on the assembly of the installation describes being inside the operating Muscle as "odd combination of celestial serenity and cybernetic beauty, with its exposed pneumatic and electrical automation hardware, its tranquilizing breathing soundscape and a continuous behavioural movement that shifts from ambiguity to precision, uniform yet multiple; a perceivable dynamic and plural extension, shaped through unseen bodily interactions, of the participants on the other side."

[1] Kas Oosterhuis, *Towards a New Kind of Building: A Designers Guide for Non-Standard Architecture* (NAi Publishers, 2011).

c) Evaluation

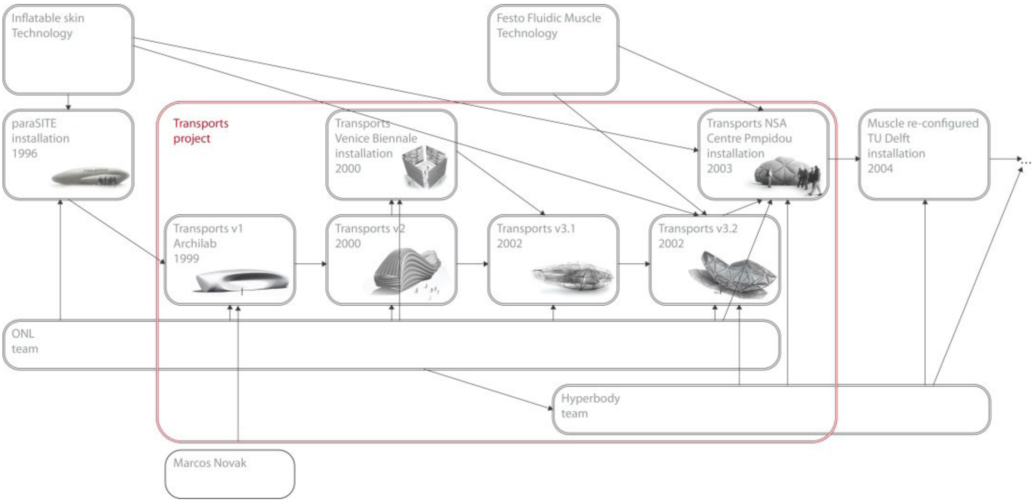

Inflatable skin Technology
Festo Fluidic Muscle Technology
paraSITE installation 1996
Transports project
Transports Venice Biennale installation 2000
Transports NSA Centre Pmpidou installation 2003
Muscle re-configured TU Delft installation 2004
...
Transports v1 Archilab 1999
Transports v2 2000
Transports v3.1 2002
Transports v3.2 2002
ONL team
Hyperbody team
Marcos Novak

Fig.28. Tracing of chronological development of projects, actors and groups of actors constituting the trans-ports as a super-project.

The trans-ports project has not been explicitly continued beyond the Muscle Trans-ports installation. However, the boundary of the project can only be seen as a convention. Similarly to how the paraSITE installation had some direct influences on the trans-ports development, trans-ports in turn has had direct and indirect influence on other projects. Most notably the muscle trans-ports installation has been donated to TU Delft's Hyperbody group, and initiated a series of "muscle re-configured" projects[1]. Every following semester students of Hyperbody would design and build a new installation reusing the control mechanisms, sensors and actuators of the original muscle.

2.3. Discussion

Tracing the development of two projects the Cockpit and Sound Barrier, and Trans-ports reveals complementary observations. In Cockpit and Sound Barrier the number of project components steadily increases and differentiates throughout the project development process. The use of mass-customization and parametric design allow to treat every component as an individual entity and permit the entire design process to be highly adaptive throughout its iterations. This delivers a distributed and parallel, rather than linear design process. Involvement of various experts is integrated in the project cycles. However, the adaptation ends upon design process completion and initiation of construction. Chosen technique does not account for any further modification or alteration of the buildings. Even initially proposed dynamic lighting solution eventually has not been realised.

In trans-ports a different design process can be observed. The project follows one leading concept; however over time several distinct designs appear where each of those could be seen as a project in its own right. This can be especially observed when muscle trans-ports installation is often referred to in articles without relation to the more general trans-ports vision.

[1] Kas Oosterhuis and Nimish Biloria, 'Interactions with Proactive Architectural Spaces: The Muscle Projects', *Commun. ACM* 51, no. 6 (June 2008): 70–78.

As in Cockpit and Sound Barrier there is a distinct start and end to the design process, in Trans-ports, the start and end are fuzzily defined. Influence of earlier projects such as paraSITE clearly played a role in idea formation, on the other hand the project continued to have a conceptual and technological influence on most later interactive installations built at Hyperbody, which could be seen as next versions or bodies in the trans-ports network.

Despite a different development track, both projects ultimately reach certain similarity in component discretization. Trans-ports starts as a continuous structure, yet eventually Muscle trans-ports is a structure built of many discrete components. The distributed parametric logic of operation is similar in both projects, with the main difference that Muscle trans-ports continues to operate and adjust parameters of its components after the realisation.

Muscle's behaviour is dynamic and through emergent processes also highly unpredictable. Nevertheless this behaviour remains highly constrained and adaptation cannot be extended beyond the prescribed ranges of movement of components. It remains an open question as of how the adaptation that could be observed between different versions of trans-ports could be continued during and after project's realisation.

The distributed behaviour of muscle trans-ports is not inherent in its structure, but simulated on its central control unit. For this reason the system has limited scalability, as for larger structures more air tubing, control cables and processing power of the control unit would become a serious bottleneck.

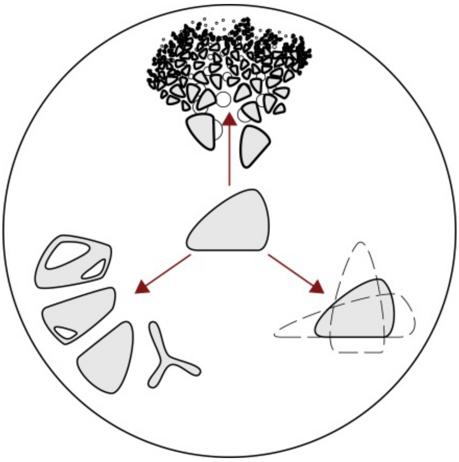


Fig.29. Ideogram: multiplication, diversification and adaptation of components as three main challenges for interactive architectural design processes.

The interaction between users and the structure has been highly evaluated by the users, and created an illusion of muscle trans-ports being "alive". However, in fact no true learning occurred in the interaction process, making the interactivity limited. In interaction scenarios beyond the time-limited exhibition, more functionally meaningful interactions would be required, of which the employed system was not capable.

The chosen technical solution has served well for the exhibition scenario; however it has also proven impractical for using the interior of the space due to air-tightness requirement.

Analysed processes point towards the main difficulty of designing complex interactive adaptive architectural systems; "how to organise development of such systems, so that their adaptation can be continued throughout their inhabitation?". Both Cockpit and Sound Barrier and Trans-ports show that discretisation and individualisation of project components in terms of their materiality and behaviour is the essential first step.

Conclusions summary:

- Working with a high number of individualised components increases project efficiency without significant increase of costs, while reducing assembly time.
- Building components are adaptive during the parametric design process, while ultimately they are manufactured as static and non-adaptive
- Adding kinetic actuation to building components shows the possibility of maintaining the substantial amount of adaptation after project's realisation
- Architectural expression and spatial qualities can be achieved by individual differentiation of every building component
- Both parametric design and interactive architectural installations are characterised by rapid iteration through the form generating process
- Form generating process can be distributed
- Project tracings show increased efficiency in horizontal team hierarchy and parallel workflows permitted by use integrated design systems
- Frequent iterations and integrated design systems allow direct involvement of clients or users in the process, interacting with virtual or realised buildings

3. Activation and prototyping

Challenges:

- Interactive architectural adaptations require dynamic building components with a wide range of spatial affordances.
- Few standard solutions exist for such building components.
- Focus is required on building "membranes" separating and differentiating architectural spaces form each other and changing degree and kind of separation.

As observed in previous section, the logical bridge between Cockpit and Soundbarier, and Muscle Transports projects would have been the usage of dynamically transformable building components, which could retain flexibility after project's realisation. Such approach would allow combining the complexity and scale of Cockpit and Sound Barrier with flexibility and autonomy of Muscle Trans-ports, while providing more adaptability of the resulting system.

What's more, complex buildings require a higher specialisation of components, where a wide range of possible spatial affordances should be potentially dynamically formed in the process of interaction between building components and its users. Dynamic building components are rarely encountered in buildings, usually in form of active doors or windows, light and visibility control (window blinds, lighting), and transportation (elevators, escalators). Clearly, many more dynamic spatial adaptations can be imagined.

Following the discussion from chapter III., architectural components can have two primary roles, which can be adapted in the process of building operation. They a) define the built space and its affordances (e.g. size of space, its qualities, provide facilities required for certain activities) b) They provide connection or separation between spaces in respect to various aspects or connectives, such as accessibility, visibility, acoustic, aesthetic and many more.

Respectively, it can be said that architectural components form "membranes", that on one hand differentiate spaces from each other and regulate flow of matter and energy between them, on the other hand they also define the capabilities of the inner working of the space they surround.

The following points trace a selection of projects where novel ways of creation of such membranes were designed and prototypes, with the ambition to serve as spatial "building blocks" of complex interactive architecture.

3.1. Muscle Façade

The Muscle Façade was a design studio project taught by the author in 2006 in the course that was part of the undergraduate bachelor 6 framework at TU Delft, faculty of Architecture. It was a sixth project in the "muscle" series[1], of which earlier projects included Muscle NSA, Muscle Body[2] and Bamboostic[3] projects. All Muscle projects have followed similar design logics. However, in the muscle Façade[4] project more explicit attention was given to the multi-agent formation and operation of the developed system.

a) Outset

The brief for the Muscle Façade project was to design a façade for a generic office building in an urban setting. The assignment was to design and build a fully operational fragment of the façade in full scale. The façade had to answer to several problems. There was a requirement for variable transparency dependent on the weather, time of the day, as well as activities in sections of the building. There was a need to communities the activities inside the building to the outside passers-by. There was also a need to alter the size of office spaces based on activities inside and climate factors. Eventually, other functionalities of the façade were encouraged.

The façade was to be considered from the start as an assembly of interconnected components. Its behaviour had to be defined in relation to inhabitants and other internal factors of the hypothetical building it was to be placed on and humans and environmental factors on the outside of the building. The student group consisted of five undergraduate students and the part-time course lasted for 10 weeks.

b) Process

From the start of the process the façade was considered to be a system of components, forming a volumetric boundary between the interior and exterior space of the building. During the initial two weeks of the course students came up with diverse ideas of what a dynamic façade component could be, how it could be made and how components could be assembled together. Among the ideas were spherical cocoons which could be entered from the inside and outside, various kinds of flexible panels and inflatable semi-transparent cells wrapped in fluidic muscle actuators and embedded with lights. Among these ideas, the last one was selected by students as the most feasible to be developed further, as there was not enough time to develop assemblies of more than kind of components. As a result first sketches of the assembly shown an array of inflatable spindle-shaped cell elements wrapped in a three-dimensional network of Festo air-muscle actuators, together forming a membrane

[1] Biloria and Oosterhuis, 'Envisioning the RESPONSIVE Milieu'.

[2] Max Cohen da Lara and Hans Hubers, 'Muscle Body', in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012), 409–415.

[3] Sander Korebrits, 'Muscle Bamboostic', in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012), 416–419.

[4] Tomasz Jaskiewicz, 'Dynamic Design Matter[s]: Practical Considerations for Interactive Architecture' (presented at the First international conference on critical digital: What Matters(s)?, Harvard University Graduate School of Design, Cambridge (USA), 2008).

inherently deformable in three dimensions. Additionally, a future possibility was envisioned to allow some of the cell insides to be accessible for various uses, e.g. as a solitary space for relaxation.

Once design vision has become established, the development of the actual process consisted of two interwoven tracks. One track was experimentation with specific technical solutions for building up the installation parts. The second track was to immediately construct the behaviour of a larger assembly of components and to design a system for their actuation.

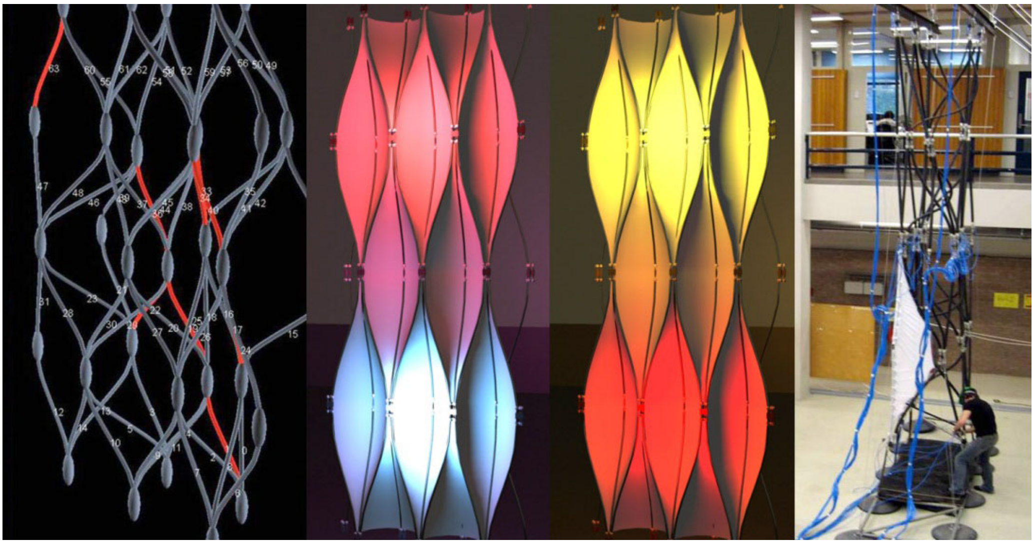


Fig.30. Muscle façade project, virtual models and test prototype

The author helped students in building a the virtual system where all façade elements were virtually placed. The components included the inflatable cells and wrapped around them fluidic muscles. Subsequently local behaviours of these components were constructed and complex patterns of muscle contractions and extensions were tested in order to understand the possibilities of deformation of the entire façade assembly. At the same time, the physical components were built as the counterpart to the virtual ones. The building of these components was iterative. First ideas were sketched, prototyped, tested and altered again.

Ultimately, the physical model became connected to the virtual one, enabling the physical actuators at the same instance that the virtual ones would be triggered. The assembly of the virtual agents has been running at a 5 times higher speed than the actual physical ones installation, thus it would always represent a projection to the near future of the installation's configuration, allowing the operator to counteract undesirable behaviours before their actual occurrence.

The developed system operated thus as a system of virtual and physical agents interacting with each other. Consequently, the behaviour of the entire system was tested and a number of operation scenarios were explored, including further addition of light sources to the inflated façade components.

c) Evaluation

The Muscle Façade shows how a virtually developed system of agents can be extended with a set of material, actual agents. In projects such as muscle façade it is impossible to predict how the system is going to operate before physically building it. The approach of "experiential prototyping" when the actual building or its part is built once being designed allows to instantly test, validate and develop further the complex behaviour of the system as a whole.

3.2. Muscle Space

Muscle façade project demonstrated the development of the multi-agent system consisting of virtual building component agents and directly corresponding to them, dynamic physical agents. The Muscle Space[1] project was the next studio in the Muscle series and was taught by the author in collaboration by the author and the fellow PhD researcher, Christian Friedrich. Seven bachelor 6 students participated in the project. The goal of the Muscle Space project was to develop a more comprehensive system, where visitors to the built installation would play a greater role in its development and where the delivered spatial organisation would be more affected.

a) Outset

The brief given to the students was to develop a dynamic spatial configuration of architectural surfaces that would provide a transition between spaces of various functionalities and that would at the same time deliver a dynamic architectural space in itself. The task communicated to the students was to develop an interactive system involving the interplay between humans and space, where no clear control relationship would have been established. Conditions had to be created where the space would guide users, as well as, in other cases where the space would be guided by its human visitors.

b) Process

Students worked as one group, in which each person had to assume responsibility for one of the following aspects of the project: structure, actuators, materials, form design, sensors, interaction, sound and light. Throughout the design process, the roles in the team would partly blur, and begin to overlap yet the basic division of responsibilities has been maintained throughout the entire project.

The initial phase of the studio concentrated on brainstorming, skill-building with small-scale prototypes and formation of the first ideas. From the start of the project, students worked directly by prototyping their project in the physical space. Teachers participated in the process as experts, with less time involvement and increased decisive power, nevertheless the organisation remained horizontal.

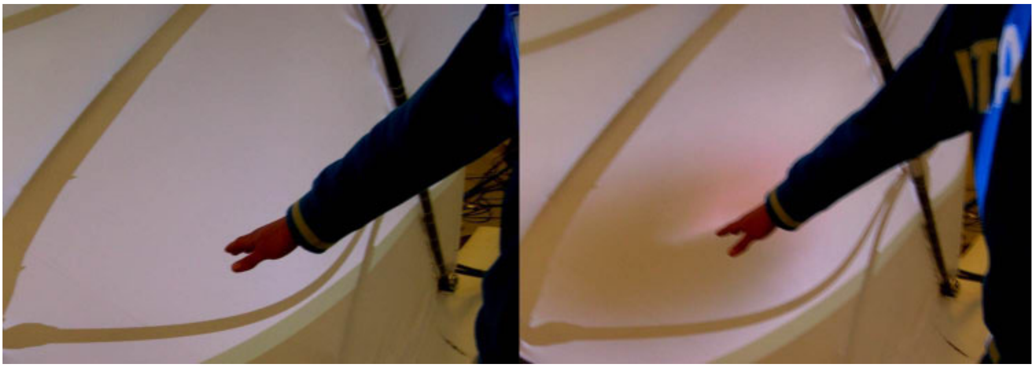


Fig.31. Interaction between a visitor and projected pattern

Initial idea for the project development was to create a system of diverse space-defining objects. Eventually, for feasibility reasons, the concept was reduced to an assembly of two flexible walls framing the space in between them and connecting two spaces at its ends. The

[1] Tomasz Jaskiewicz and Christian Friedrich, ‘Muscle Space’, in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012), 420–424.

kinetic walls were envisioned to be capable of extensive bending in- and outwards. Not only the effects of a chosen system's actions on people, but also its actual structural performance would have been substantially more difficult to analyse without building the prototype.

The modularity of the project was less straight-forward than in other to this point presented projects. The walls consisted of flexible struts, connected with hinging joints to form a three dimensional surface, capable of scissor-like extension and contraction occurring in parallel with its three-dimensional deformation. The actuation of the surfaces was achieved by fluidic muscles diagonally placed between hinging joints. Each of the struts, joints, and muscles was initially considered to be an autonomous component. Once the first assembly of such elements has been tested, it became clear that the whole assembly can be treated as one agent. The deformations of the entire assembly were complex. One assembly contained 9 fluidic muscles, thus there were 512 possible configurations of fully extended and extracted muscles, and endless in-between configurations. However through geometric calculations based on empirical testing of deformations, an approximation of the deformation could have been calculated. As a result, the two wall assemblies could be considered as two autonomous agents, nesting approximately a hundred individual components within.

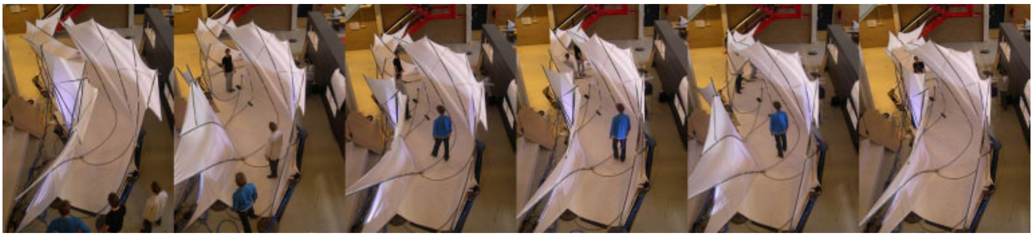


Fig.32. Transformation of muscle space over time

Based on two flexible walls, the design concept has been developed further and spatial layout has been developed, by moving and adjusting the actual walls on site. Once the elements were standing, a wide range of behaviours could be developed and tested. Initially students used each other as test subjects, eventually more testers were invited to further analyse reactions and subsequently, possible types of feedback to base the adaptive installation's behaviour could be investigated. The ultimate behaviour was effectuated by kinetic deformations of installation walls, light projection images on the walls and sound from a distributed set of 6 speakers. The behaviour consisted of a rhythmic deformation of the entire installation, accelerating and at the same time decreasing its magnitude locally while nearby presence of people was sensed through floor pressure mats and proximity sensors embedded in walls. In this way, people passing through installation space were slowed down as if inspected by the installation "organism" and gradually encouraged to move on through its space, by the rhythmically reoccurring opening of the structure. The effect was increased by a ripple like pattern projected on the installation walls accentuating "fields" surrounding the visitors, as well as the ambient, distorted buzz of the spatial installation soundscape. The final project has been deployed in two iterations. First test iteration took place at TU Delft, where the installation was publicly exposed while its behaviour was being adjusted. The second step was a 4-week exhibition at TOP Delft gallery, open to the general public.

c) Evaluation

The project has been from the start developed hands on. Teachers and students working together could have been also seen as inhabitants of the built space. The design of components and behaviours for them progressed iteratively and there was no predetermined order to how it was performed. The space of the “bouwput” area at the faculty of Architecture created an ideal context, as the Muscle Space laid between two corridor spaces, providing a connection between them.

The resulting assembly of agents was truly heterogenous. It consisted of two main wall-surface agents, equipped with pressure mats to detect presence of nearby humans. These two main agents were built up of nearly 200 “sub-agent”, other components of building up the wall membranes. However, the virtual agents played an equally important role in the system. Wave agents were generated in direct response to activity of walls and passers-by and projected onto the wall surfaces. Sound agents followed a similar pattern of emergence, and were brought into the actual world through an immersive sound installation. It was an emergent assembly of autonomously behaving, interacting with each other and with installation visitors agents, that created the rich spatial experience. The development of the constructed microcosm was not controllable, nor was it controlling humans. It emerged through the process of interaction between humans and things forming one ecosystem[1].

3.3. Interactive portals

The interactive portals project took place in the framework of the Hyperbody vertical studio, where students from different study levels were joined by one assignment. The course was also a first test for collaboration with the ID-StudioLab department at the Industrial Design faculty of the TU Delft, where other two student groups received the same assignment. The premise of the course the methodological line for development of interactive architectural spaces, but to propose a challenge that would introduce a higher degree of complexity to the developed projects[2,3.]

a) Outset

The brief of the studio was to build a network of interactive “portals”, consisting of individual elements capable of local interactions with their surroundings and at the same time acting and interacting with other portals of the network. The idea of a portal may be interpreted in many ways. Historically, in both western and eastern architectural cultures, portals have always existed as very important and symbolic structures. Despite diverse forms being used to mark them; in all cases they have shared a common feature of creating a transition from one space to another. Futuristic visions of portals show us that there are many more ways of interpreting this idea. Examples of portals from Star Trek or other science fiction movies trigger our imagination as of what may be possible with the use of future technologies, including ideas of stepping into a portal and transporting ourselves physically or virtually to even the most distant locations. Even though materializing such a vision is still not within our reach, the role of the assignment was to stimulate students to think out-of-the-box abd diverge their initial ideas.

[1] Sebastian Moffatt and Niklaus Kohler, ‘Conceptualizing the Built Environment as a Social–ecological System’, *Building Research & Information* 36, no. 3 (2008): 248.

[2] T. Jaskiewicz and N. Schueler, ‘Prototype Presentation: Interactive Portals Hyperbody’, in *The Architectural Annual 2007-2008, Delft University of Technology*, ed. L. Calabrese et al. (Delft, 2009), 48–51.

[3] Tomasz Jaskiewicz, ‘“iPortals” as a Case Study Pre-Prototype of an Evolving Network of Interactive Spatial Components’, in *Proceedings of the 28th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)* (Minneapolis, 2008).

b) The leafs portal

The starting point for the “leafs” group was researching in a critical way various spatial typologies of portals and their possible behaviour in time. Finally they reduced their outcome to two conceptual and functionally opposite ways of interpreting the notion of a portal: the physical and the virtual portal.

The physical portal has been defined in a traditional way, related to conceiving a portal as a doorway which establishes a connection between two spaces. To get to the other space and to access the information or physical objects within, people have to physically pass through the portal. In this case the goal is static and space has a more dynamic role, transforming itself as the person passes through it. The virtual portal works in the opposite way. While the information that the user is looking for is found, gathered, and brought to him through the portal, the whole space in which the person is located remains static. From this analysis, the group derived their design concept to create a portal capable of giving the impression of the flux of information flowing through it, from the user to the network and vice versa, related to the movement of the people in the space.

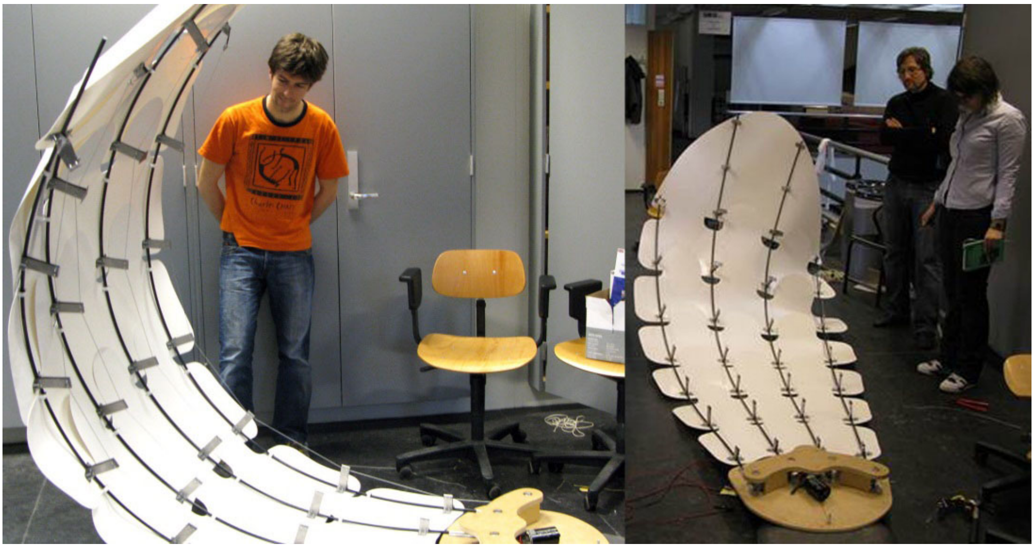
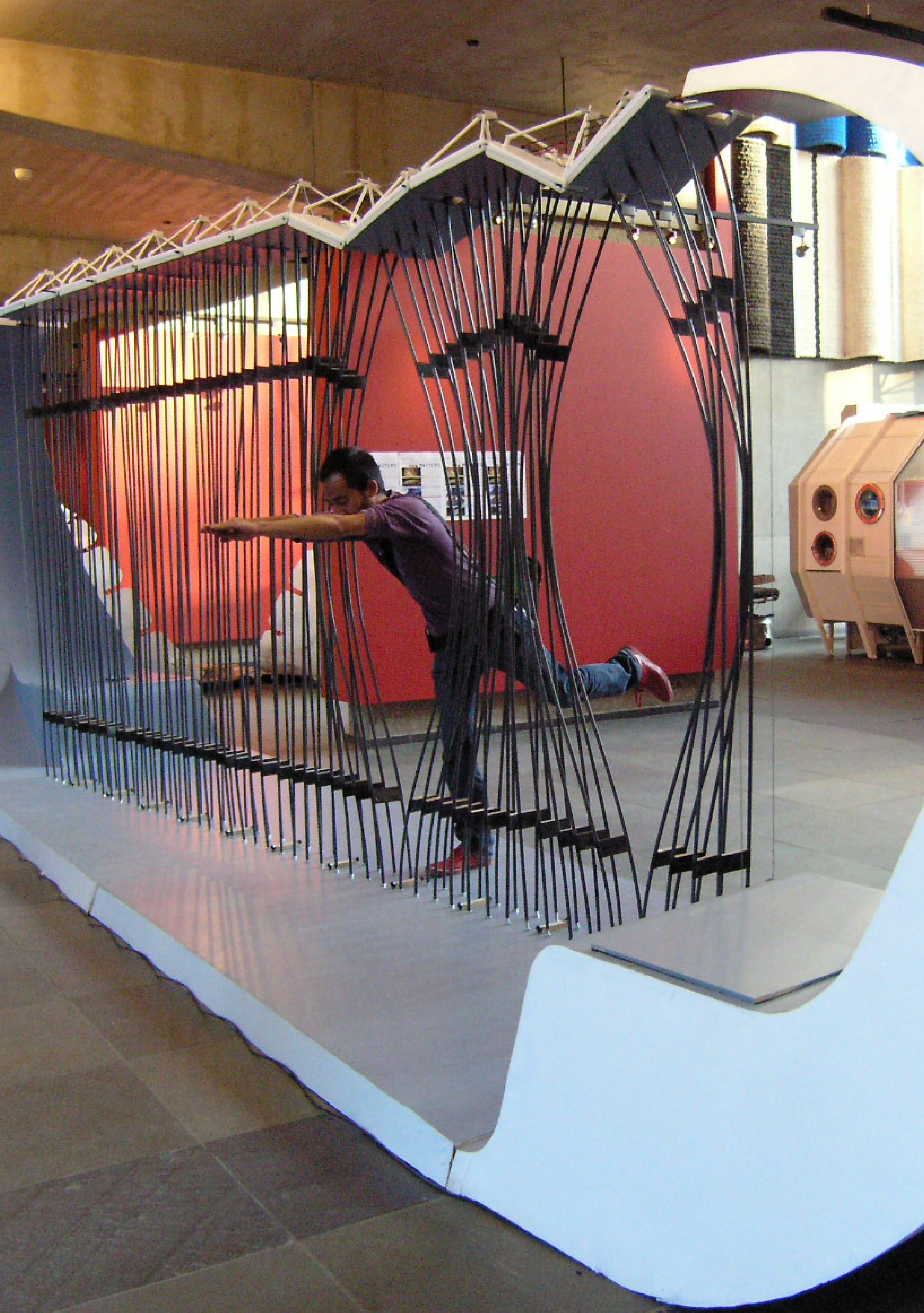


Fig.33. Leaf portal prototype

The first idea of the group was to build an interactive surface that could curl up from the ground to form an interactive landscape. Same quickly as it appears, it could deform back into a flat surface again. Very soon this idea evolved, from the design of one surface to a group of surfaces that are connected to operate as one installation.

Furthermore, the installation setup was augmented with an interactive behaviour and standard rules for operation. Three different behaviour states of the leafs were defined and related to a respond to different action patterns of the people. The stand-by status happens, when no users are present in the surrounding, in which the surface elements remain stretched flat on the ground. The passage status occurs, when users pass without stopping through the space in which the elements are situated. In this state the surfaces start to play simulated or real time sounds coming from the other portals and begin to move accordingly. The third, enclosure status, begins when users stop at a determined distance. In this state they are allowed to intervene with the data flow of the network of portals by moving in the space of the portal and changing the sound played by the surfaces.



c) The skin portal

The second group started from the simple idea. They have asked themselves why should there be a space if it is not needed. They defined a portal as an entrance or a passage that gives a meaning to the act of being in the space of the portal. Their installation can be understood as a segment of a building, preferably the skin. The group argued that the skin of the building of the future could have more power than only a passive wall with a doors and windows. The challenge of the skin portal is to add more responsibilities and possibilities to walls of a building.

Building "membranes" should communicate a meaning or a mood in a dynamic way and react instantly to their surroundings. In this way an interactive wall should not only create openings to entrants but maybe even select them according to certain parameters and invite them into the building. The other way around would also be possible. Such wall could change the interior space of the building accordingly to the people that enter.

Eventually the group has decided to build the installation as a lounge element that would resemble a part of such a future building skin. A curtain of pvc tubes creates openings for those that pass through according to their proximity and position along the "curtain". Additionally, it also creates spontaneous openings based on information coming from other portals, simulating a virtual guest.

Just like the first group, this group also had to put enormous effort into materializing their installation/ With the help of the inHolland composite lab the group made light weight composite roof panels that connected with cables and motors to enable the opening mechanism. Sensors, lights and many other technical gadgets have been used to allow designed interaction.

d) ID-StudioLab portals

In parallel to the two above described projects, the same assignment was given to two groups of Industrial Design students. Because of organisational constraints industrial design students had less time to develop their projects. However their interpretation of the topic provided a yet another interpretation of the challenge.

The "jealous portal" concept was to create a structure that would be a centrally placed element, separating spaces around it by lowering and raising its outstretching branches. It not only creates spontaneous divisions of spaces surrounding it, but also affects the atmosphere of its environment by actively engaging passers-by in playful experiences. Its operates by locally communicating and entertaining its visitors, but also senses amount of activity present in other portals in the network and tries to attract their users to itself. In this way, it develops a "jealous" behaviour, trying to steal user attention from other surrounding spaces.

The "bubble pods portal" design approached the notion of a portal in a similar way to the leaves project. Although the project was destined to operate in a more ambient manner, it provided a flexible spatial setup and involved its users in a playful spatial interaction, allowing manual relocation of its man-sized elements. When sensing motion or proximity, elements act accordingly with light and sound feedback, encouraging to be moved or discouraging from being touched.

The two portal projects developed at the faculty of industrial design, put even more emphasis on user involvement. The behaviour of created prototypes was thoroughly tested and fine-tuned based on feedback from a user test group

Img. 6. Left: iPortals, Curtain Portal project.

e) Portals integration

The initial plan was to connect four developed portals into a network of installations and integrate them to a larger scale system. The ambition was to demonstrate how a group of inhabitants-designers can construct a complex spatial environment through local construction of components, assemblage of these components into a working spatial module and eventually interconnection of such modules into a larger scale, more heterogeneous and more complex architectural system. The fire of the faculty of Architecture in May 2008 has prematurely terminated the projects and the last step of the process could not have been performed.

f) Evaluation

The projects took different paths towards interpretation of the assignment and the developed building components took different approaches. In all cases students started building and testing their components very early in the development process. Students of industrial design, approached the components as "loose" objects. One such object was developed in the "jealous portal" project, a "swarm" of such objects was developed in the "bubble pods" project. Clearly, such objects can modify the qualities and affordance of an architectural spaces, but only in limited ways can they constitute the spatial boundaries. The "curtain" project developed the opposite. The individual agents – sections of the curtain were tightly integrated and could not perform in separation. The created surface was a typical "building block" of a building extended with an unprecedented ability to freely modulate its permeability. The "leafs" project delivered an in-between system. Individual "leaf" surfaces could operate autonomously, but through spatial interaction, they were able to form larger spatial surfaces.

In all projects, the interaction among created objects and humans developed gradually from one-to-one interactions to systems involving many humans and many objects interacting with each other. It has been observed that such gradual development of interactions leads to robust systems, where stable, robust state of the system can be modulated and prevented from turning into a chaotic one.

3.4. iLounge

In 2009 the minor programme named "Interactive Environments" was established as a reinitiated combined education and research vehicle at the Delft University of Technology. This programme was formed as collaboration between the Hyperbody from the faculty of Architecture and the ID-StudioLab from the faculty of Industrial Design and Engineering. In the first year it had been coordinated, co-organized, and co-taught by the author. The main purpose of the programme was to create facilities and organisation where interdisciplinary researchers, practitioners and students could be brought together to work on the ideas of embedding interaction in architecture. This setup has provided a test field for verifying the applicability of various design methods for design and creation of architectural interactive systems, inherently tied to building full-scale experiential prototypes of developed interactive architectural systems. The first project established in the Interactive Environments programme was the iLounge[1].

[1] T. Jaskiewicz, A. van der Helm, and W. Aprile, 'Creative Approach to the Design and Prototyping of Experimental Smart Spaces, Case Studies from the Interactive Environments Minor', in *Smart Spaces and Next Generation Wired/ Wireless Networking*, vol. 6294, Lecture Notes in Computer Science (presented at the Third Conference on Smart Spaces, ruSMART 2010, St. Petersburg, Russia: Springer, 2010).

a) Outset

The posed challenge in the iLounge project was to design a small scale pavilion, "small enough to be placed inside another building and large enough to accommodate many users and uses, which could be fully enclosed or left entirely open". It was given a generic function of a lounge. In this context a lounge meant "the kind of space where people can take a break and relax, while also allowing them to participate in other social and professional activities; solitarily, with other lounge-users or with remotely connected partners. It is also pro-active; not only continuously adapting to changing users and their demands, but also anticipating and actively influencing these demands, sometimes in surprising and unexpected, at other times in ambient and barely noticeable ways."

The project had been planned as a semester long endeavour. It has been divided into two parts. In the first two months students have been engaged in various skill- and knowledge building assignments, while teachers have been closely monitoring the students, as well as exchanging knowledge and approaches between each other. Throughout the whole semester, five teachers have been continuously involved in the project; their backgrounds included architecture, industrial design, computer science, interaction design, new media and digital music. Throughout the course many other specialists were invited as lecturers, workshop teachers and consultants.

Students participating in the programme had diverse backgrounds. Of 21 students, one third were students of architecture, one third of industrial design and engineering and the remaining ones were majoring at faculties including aerospace engineering, computer science, applied physics and mechanical engineering. This melting pot of experiences allowed to creatively rethink the approach to design along its execution in their main assignment.

b) Process

In the first part of the semester students were engaged in a series of short design and prototype assignments, where they had to design and build simple interactive objects for specific functional or experiential purposes. Gradually, the scale of their design was shifted from individual objects to networks of objects interacting with each other and with their users. In parallel to these assignments, students were also conceptually designing an interactive lounge environment, based on ideas explored in their quick prototypes.

The first half of the semester has ended with a general design of three interactive lounge installations, each prepared and presented by a different group of students, where individual students were responsible for different parts of the system. Following this, in the second half of the semester the students with assistance of all teachers have embarked upon realizing their design visions.

It has been empirically proven very early in the process that typical architectural design approach is not suitable for given interactive design assignment. Even though the function of architectural spaces that students had to create has been defined as "lounge", required interactivity of that space made it impossible to design fixed pavilion forms that would allow for maintaining a continuous reciprocal interaction between users and the pavilion. Students have proceeded by defining lists of user activities that their pavilions should accommodate. Following that, lists of specific spatial affordances were created. To help with achieving design coherence, each group defined a concept that would in an abstract way guide the experience that each installation was trying to provide for their users. These mottos, later on in the process put aside, included "parallel universe", "artificial nature" and "immersive ecology".

A big change could have been clearly observed throughout the skill-building workshops in the way students were thinking and talking about their projects. In the beginning of the course their descriptions and ideas about their design's behaviour were rather naively approaching stated problems, even the most technologically demanding ones were not going beyond the

reactive manner of operation that could be entirely described using a series of if... then... statements. Gradually the students have gotten to understand the systemic nature of their designs. An important trigger for this understanding was a three day workshop with Ruairi Glynn[1], where the students had to design small, light-sensitive robotic "creatures" and build a "performative ecology" out of all robots. The descriptions of projects became more oriented towards explaining the experiences of users in their space and attempting to define those experiences through a multitude of localized interactions between users and the designed installation components, among the components and, what's important, also among the users. In this way all designs have gradually drifted towards becoming ecologies in which both users and installation components were actors. The step in which student projects have shifted from top-down and centralized ideas of interaction to the understanding of their projects as complex, adaptive ecologies has been in all three cases a noticeable breakthrough. The profoundness of considered interactions and resulting formal complexity of projects has quickly increased and the projects presented at the mid-term review were very satisfactory.

The project of the group of the students[2] who named themselves the "Odyssey" team, envisioned a setup where the space of an installations has a clearly differentiated outside and inside character, creating a surreal feeling of an immersive, continuous space on the inside and a crude, mechanical form on the outside of the pavilion. The behaviour of the pavilion is organized by the concept of virtual waves emitted by people entering the space of the pavilion. These waves would be received by the kinetically actuated structure of the installation, distributed ambient lighting elements and 40 small actuators embedded in the floor of the space. The centrally placed, large sitting cushion connected to the ceiling of the pavilion would act as a heart of the system, collecting signals from all dynamic installation components and based and translating their behaviour into continuously generated sound, while occasionally creating an energy explosion and sending out its own wave. Yet to be verified upon the completion of the project is whether as envisioned, this behaviour will lead to the perception of the space as intelligent and whether through repeated interactions users will learn how to meaningfully interact with the Odyssey installation.

The project of the second group[3], self-proclaimed as "GEN". Includes a static canopy structure, leaf-resembling kinetically actuated vertical spatial separators and a complex pneumatic floor "landscape". The group intended to create an installation that would maintain a non-repetitive, yet cyclical behaviour in which it will constantly try to create conditions that it perceives as most favourable by its occupants. By tracking the faces of people inside and outside the installation area it maintains a continuous track of occupant activity. Components of the installations that are in use (e.g. sat on) are seen as successful and gain virtual "kudos". In the process they can trade kudos among each other to overrule the behaviour of their neighbours. It is expected that such "kudo economy" will lead to a continuous adaptation of the entire system. Next to this part of the behaviour, the installation will include small dose of random factors, to make sure that new behavioural patterns also evolve in the process.

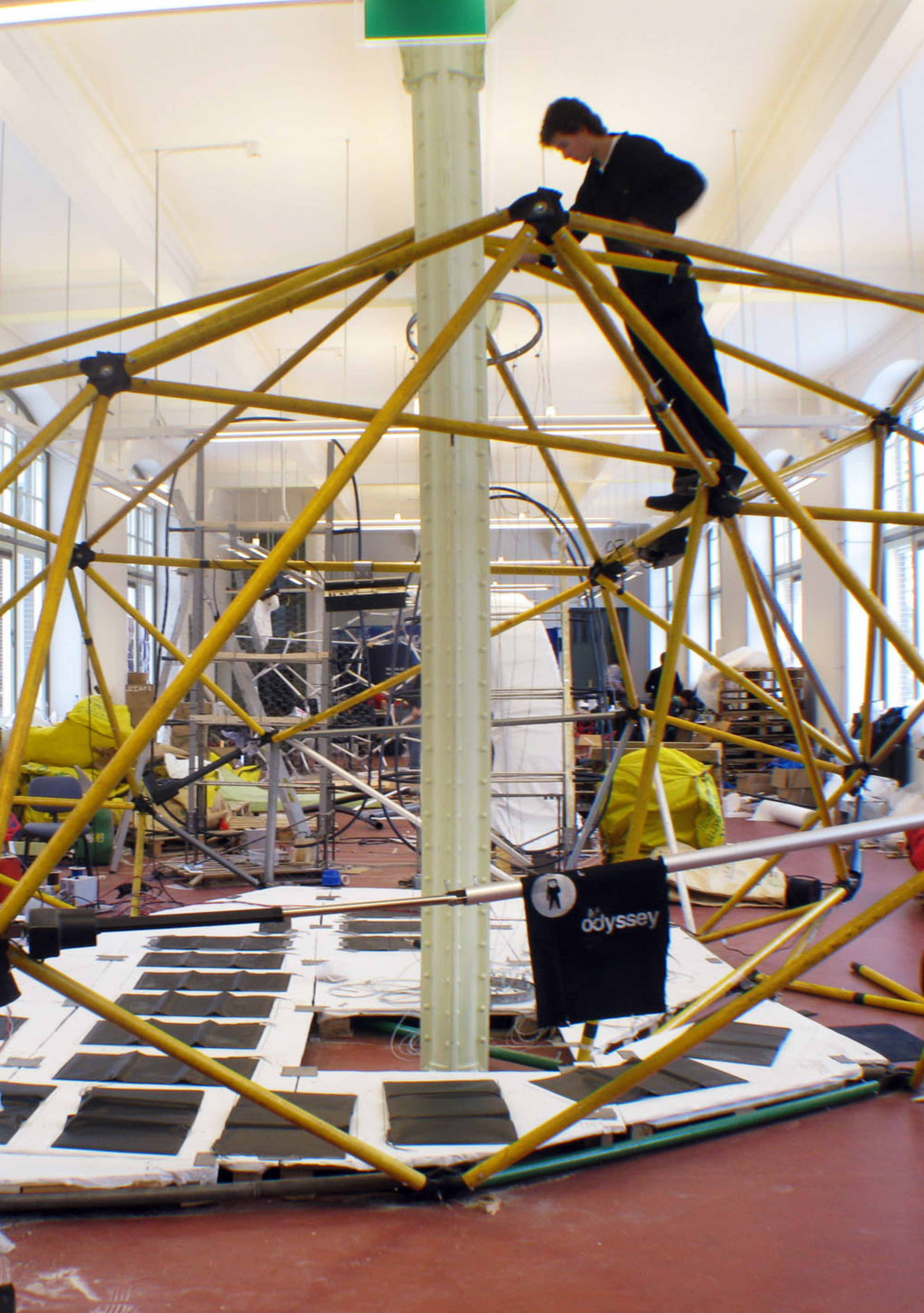
The third project codenamed "sCAPE" aimed at creation of the pavilion being a fully distributed system. Their main concept is the idea of "LEDwork" elements that can be freely connected to each other to create a three dimensional spatial network. Each element has its own simple behaviour and can locally exchange information with other elements it is physically connected to. The basic elements are light-emitting ones, free to be moved around. All other parts of the installations are mutations of those. The structure of the pavilion is built out of

[1] Ruairi Glynn is a lecturer at the Bartlett School of Architecture and Central Saint Martin college in London, he runs the online blog interactivearchitecture.org

[2] Students in the Odyssey group are: Govert Flint, Lieke Kraan, Bob Groeneveld, Jesse Timmermans, Merijn Pen, Thomas van Oekelen, Melisa Garza Vales

[3] See appendix 1 for project team members and credits

Img. 7. Left: Interactive Environments Minor, sCAPE project


odyssey

struts and nodes, where each node is regarded as a LEDwork node, fixed in place and can be connected to all mobile nodes. Other fixed nodes include additional lighting elements, sound emitting components and more. The intention is to create a dynamic ecosystem in which the visitor to the installation can completely immerse, while becoming a part of it him- or herself.

The second part of the semester has been entirely devoted to building the installations in full-scale. It has been explicitly stated that projects upon which the designs were based were not to be seen as final blueprints, but to the contrary, they were supposed to be modified and evolve throughout the entire building and testing process.

As the first quarter could be described as an iterative skill building period, the second quarter has been devoted to iterative building of complete design prototypes. Students had to fit their installations in a tight budget, thus they had to experiment with various materials, many of which have been recycled waste or leftovers from industrial production. Their designs had to be constantly adjusted based on structural and aesthetic properties of the used materials. For this, invaluable support was provided by quest teachers Denis Oudendijk and Jan Korbes from the Refunc[1] group. The interactions have been tested and adjusted on the fly, based on on-going experiments with embedding sensors, actuators and effectors into the built structures. To facilitate the prototyping process, the groups were further subdivided and each subgroup of two or three students was made responsible for a different part of the installation. In this way, installation elements were prototyped individually and gradually integrated into a bigger system. The process of building the installations was thus interactive in itself. By interacting with the materials and component prototypes, students were adapting their designs and adjusting what was being built. In this way many creative solutions were found for difficult problems.

c) Evaluation

In the iLounge project and unprecedented project setup has been tested. Involved students were coming from various background disciplines. Many of them had little or no experience in creative design, while others had little or no experience in working with technology. Therefore the course had to include an extended didactical set of modules. The diversity among students created significant frictions among them and delayed the initial development process. However, over time students developed understanding of each other's skills and assets and learned to distribute the work accordingly. Once the interactions between such designers with diversified expertise was stabilised, the groups focused on prototyping their projects. Initial suggestion for the students was for one person in each group to be responsible for one building component, to assemble these components within groups, and eventually to attempt to create a larger ecosystem of all 21 components. This strategy proved unattainable, as due to diverse skills, students were not able to work on their own. Instead groups focused on development of holistic interactions for three individually designed components. Within groups the projects were developed similarly to earlier presented design research case studies. The students built quick mock-ups of some components, tested interactions, developed the components further, and added more elements. Iteratively the three pavilions were developed as comprehensively working ecosystems. Eventually the pavilions were erected next to each other. Each pavilion provided a unique spatial experience. However, no direct interaction between pavilions was achieved.

Over two hundred people visited the exhibition during three weeks of its operation. The reactions of visitors were positive; in most cases it was observed by them that the transformations of the space occur non-linearly and that a form of dialogue was noticeable

[1] Jan Korbes and Denis Oudendijk, 'Refunc', accessed September 20, 2010, http://www.refunc.nl.

Img. 8. Left: Interactive Environments Minor, prototyping process

between them and the space. Installations were most often compared to “animals” or “plants”, much less frequently to “robots” or “machines”. This showed that the interaction occurred more naturally than in case of earlier experiments.

3.5. Discussion

The traced projects have explicitly shown that in design of new kinds of dynamic architectural structures, material and experiential prototyping are a necessity. Technical solutions radically influence the design of structures and with dynamic systems many problems are impossible to anticipate. What’s more, actual interactions have shown to largely differ from what was initially envisioned by students. Predesigned linear interaction scenarios rarely progressed as anticipated (e.g. behaviours of the structure designed to lure visitors inside acted as repellents). Interactions had to be consequently multiple times re-designed following user tests. As a consequence designers found themselves working “within” the designed systems, becoming a kind of “expert users”, able to extensively modify and tweak prototyped installations.

All kinds of flexible building elements proved to be more constraining than initially expected. Consequently, largest transformations occurred as aggregations of smaller transformations, rather than transformations of single components.

Scalability of designed systems has shown to be a significant bottleneck. Where prototyping a single component was an attainable task, prototyping several components has shown to exponentially increase the difficulty. Prototyping larger number of components, including combining already developed and prototyped components into larger systems proved unattainable for student-designers. Technological constraints were identified as equally or less hindering than logistical and methodological ones. It has also been observed that unpredictability of behaviours in systems consisting of many components can be destructive and dangerous to the installation and its visitors if not anticipated and if preventive mechanisms do not become embedded in designed systems.

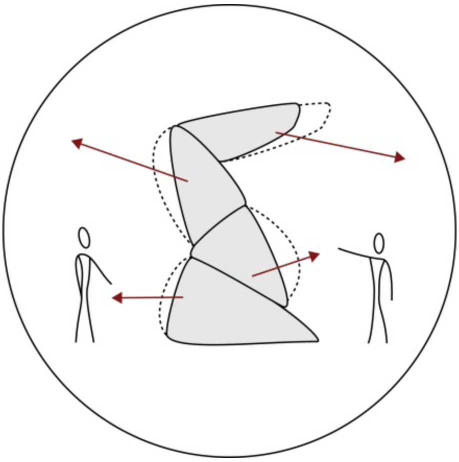


Fig.34. Ideogram: building actuation and actualisation resulting in an interactive feedback loop with users and/or designers.

Eventually, installations have not been designed for any particular context. As a result they were expected to work as autarchic systems, where actual applications would always be nested in other systems. It has been shown that placing an installation in a different setting, potentially involving a different user group would entirely change the interaction patterns.

Img. 9. Left: Interactive Environments Minor, Odyssey project

Collecting feedback from users has been a difficult task. Not having a cultural reference point has shown to be problematic for interviewed subjects, as many did not understand what to expect from the structures. This has led to the open question how to create cultural models of open affordances, in other words “affordance of changing affordance”?

Conclusions summary:

- Dynamic building systems are difficult to predict and simulate due to difficulties in determining dynamic composite element properties and complex dependencies between elements.
- Rapid prototyping proves to be an efficient way to determine dynamic building assembly behaviour and allows its iterative optimization.
- Experiential prototyping allows studying user interactions and provides immediate feedback to the design process.
- Transformation constraints require variation between elements in order to create diversity.
- In centrally controlled systems, difficulty and cost of prototyping raises exponentially with the number of components.
- In systems with larger numbers of components mutual influences between components become unpredictable and potentially destructive.
- Experiential prototypes created in laboratory cannot be comprehensively tested due to the lack of actual project context and cultural reference.

4. Multi-component formations

Challenges:

- How to progress to richer systems with more dynamic components?
- How to progress to full-building systems with comprehensive functionality?
- How to turn uncontrollability of emergent behaviour into advantage?

As discussed in section 2, one of the main challenges for further advancement of iA design lies in developing building systems with high numbers of individualised and heterogeneous components, while at the same time allowing these components dynamic adaptability and autonomous behaviour. Section 3 has shown several directions in which such components can be developed, yet there design approaches have dealt with designed installations as wholes and interactions between installations and inhabitants, rather than being explicitly concerned with interactions and relations among installation components. However, shall an installation be designed as a system of tens, hundreds or thousands of components, top down design of global behaviour and fixed interaction scenarios would no longer be possible.

In this section a selection of projects is traced, which were author’s original attempts to design building systems as aggregations of building components. Tracing the personal design processes allows to gradually explore the practical side of such approach, starting with the possibilities coming from basic assemblies of adaptive components, eventually going towards more complex systems and highly intricate emergent behaviours and qualities they can deliver.

4.1. Emergent Playground

a) Outset

The act of play is often connected to spatial affordances and often involves a number of people. The ADA installation[1] discussed in chapter III has been direct inspiration for the author. The question asked was "how could ADA's behaviour become more architectural?", not only interacting with its visitors through light and sound, but also by influencing the spatial qualities and affordances of the spaces. The true challenge for the project was though to develop an understanding of what design method would be naturally employed to design such system at an early, conceptual stage and what would be required to further develop such design into a working system.

b) Process

The design of the Emergent Playground started from the design of one tile. The concept involved adding an extending mechanism, otherwise known from automatic traffic bollards. Such mechanism allows individual tiles to rise significantly above the ground, with the height determined by the available depth beyond ground plane and mechanical constraints, but potentially exceeding 1m. Each of such tiles was envisioned to be equipped with a colour and intensity changing LED light source on its perimeter, integrated speaker and proximity and force sensors, monitoring the activity in the space above. With such design, individual tiles would be technically able to deliver multimodal interaction with its visitors. The hexagonal footprint was consequently chosen to allow tight packing of more tiles next to each other.

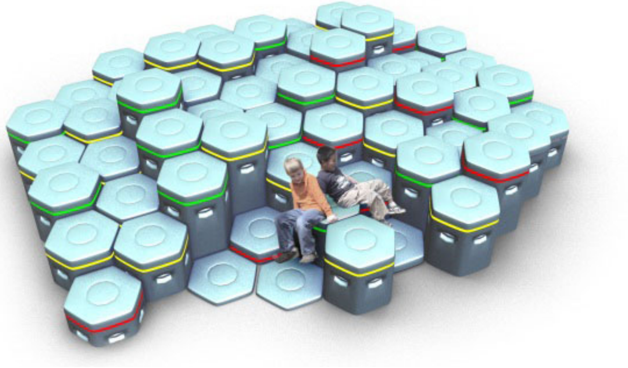


Fig.35. The Emergent Playground concept

Knowing the technical capacities of one tile, more concrete interaction ideas were imagined and conceptually sketched. An idle tile's behaviour was envisioned to gradually enhance a semi-randomised change of state, producing light patterns, sounds and changing its extension in order to attract attention of passers-by. On the other hand, too "aggressive" behaviour was likely to repel passers-by, being potentially interpreted by them as a warning message. Therefore an internal feedback loop mechanism was sketched to tune down or tune up behaviour patterns that would not lead to gaining new visitors.

In the next step an aggregation of tiles was imagined and sketched. Although several games and playful activities could be imagined involving a single tile, clearly, it is a "landscape" of dynamic tiles that provides more possibilities. Due to envisioned placement of the installation in a public space, no manual could be given to every passer-by. The interaction would have to

[1] Eng et al., 'Ada - Intelligent Space'.

be natural. On the other hand, not knowing the age group, weather or cultural background of passers-by, or even not being able to predefine their number, meant no play scenarios could be designed. Instead, the installation was envisioned to provide emergent behaviour, where sensed reactions of passers-by would either enforce or diminish certain behavioural patterns of groups of tiles. A number of scenarios were consequently imagined, such as children playing hide-and-seek, where a child hiding still would be slowly surrounded by raising tiles, or a social encounter scenario, where small groups of pedestrians would be provided with secluded sit-and-meet spots.

c) Evaluation

The project has shown the most rudimentary design process where an aggregation of active components can lead to emergent and unexpected interactions. It has also demonstrated that no prototyping or other technological means are explicitly necessary for early ideation stage.

However, the designed interactions have only remained conceptualised. Despite apparent feasibility of the interactions, no certainty could be given in respect to the actual scenarios that would actually unfold in the installation. Its eventual success as architecture would largely depend on the way in which it would have been programmed and whether intended playful engagement through emergent interactions would have been achieved. Due to the social complexity of these interactions virtual on-screen simulation would have provided very limited feedback. The actual prototype on the other hand would require fully functional tiles, where the estimated production cost would be exceeding 2000€, with higher early prototyping costs.

4.2. Bubble Lounge

a) Outset

The emergent playground components were limited in their architectural applicability to creation of dynamic architectural spaces in three dimensions. The Bubble lounge design has been initiated as an attempt to create a three dimensional transformable structure that would learn and adapt to activities of its visitors by dynamic deformation emerging from local interactions between components similar in kind to this of emergent playground.

b) Process

The Bubble Lounge was from the first sketch on envisioned as a structure built up of flexible pneumatic cushions embedded with a tensegrity inner core, where tensile elements were linear pneumatic actuators – fluidic muscles produced by Festo. Each of the cushions was envisioned to include an embedded microcontroller and a set of sensors for local awareness of external conditions and communication with the installation users. The wired connections allowed communication with neighbouring components.

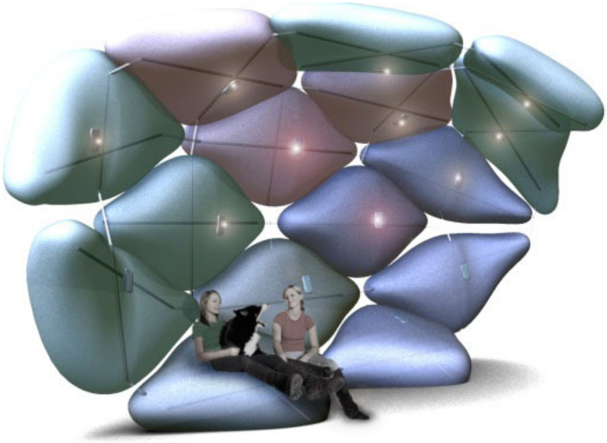


Fig.36. The Bubble Lounge concept.

In this installation not only the actual behaviour of networked components was unknown without iterative testing. Also structural integrity and overall shape of the structure were impossible to estimate without testing with the actual material. For this a prototype of three cushions was made and initially tested in order to deliver general information on components' structural performance, but also its actual shape and feel. The preliminary installation prototype has been destroyed in the fire of the Delft's Faculty of Architecture building in 2007, which has prematurely terminated the project.

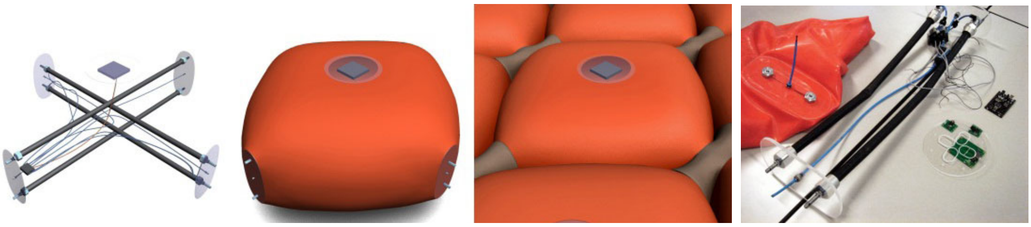


Fig.37. Cushion component design and early prototype.

c) Evaluation

The project extends the design process introduced in the emergent playground in a twofold manner. Firstly, it adds the degree of complexity and interdependence between components in the system by structural and three dimensional dependence between them. Secondly, it involves component prototype in the process.

4.3. EvoStructure

a) Outset

The EvoStructure has been initiated based on the design concept aimed at exploring another possibility of emergent multi-component systems. In Emergent Playground and Bubble Lounge, all components were identical and permanent in the system. The leading idea of EvoStructure was to introduce diversification of components and allow their replacement over time. Similarly to previous projects, in EvoStructure the aim was to explore early ideation process of one-person design.

b) Process

The EvoStructure project has been consequently sketched as an assembly of flexible nodes and struts, of which some were kinetically actuated. In this case, structural members were envisioned as changing in length, dramatically altering the form of the structure depending on given local conditions. Out of such components, any three dimensional form could be constructed.

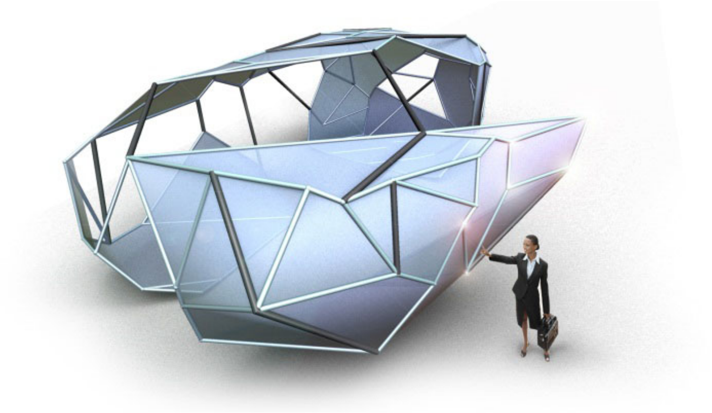


Fig.38. The EvoStructure concept

The following step in the design process has been to imagine the assembly process of the structure. Fundamental questions were "would the project be about providing users a kit of parts, or should there be an initial state pre-designed?" or "would the structure govern its own growth or decay, or would it be left to its users?" Soon it became clear, that the process of structure's development had to be designed before any other aspects of the process could be indicated. The architectural form became the last step to be delivered, where sketches were indications of a possible form the installation could take, rather than explicit designs.

Design decisions were made to pre-design a starting setup of the installation and to propose installation's fully autonomous adaptation. However without developing a working prototype and testing actual development of the structure and various programmed behaviours, no conclusive solutions could be assumed as feasible. The cost and workload of actual prototyping proved prohibitive to continue the project further.

c) Evaluation

Tracing development of the EvoStructure project has shown the inherent difference in designing pre-determined architectural system and openly developing ones. In the second case system's rules, technical constraints and complex emergent properties dependent on system component behaviours and users alike determine much of the eventual architectural qualities delivered by the system.

4.4. D|E|Form

a) Outset

The three projects discussed until now in this section have been remained in the conceptual phases. Their development has indicated that in complex adaptive architectural projects, as the number of components increases, the shape and performance of conceptually designed architecture can only be speculated upon. The d|e|form project was designed as an attempt to integrate the ideas developed in Emergent Playground, Bubble Lounge and Evo Structure while also advancing the development process of the project into virtual and physical deployment of the designed system.

b) Process

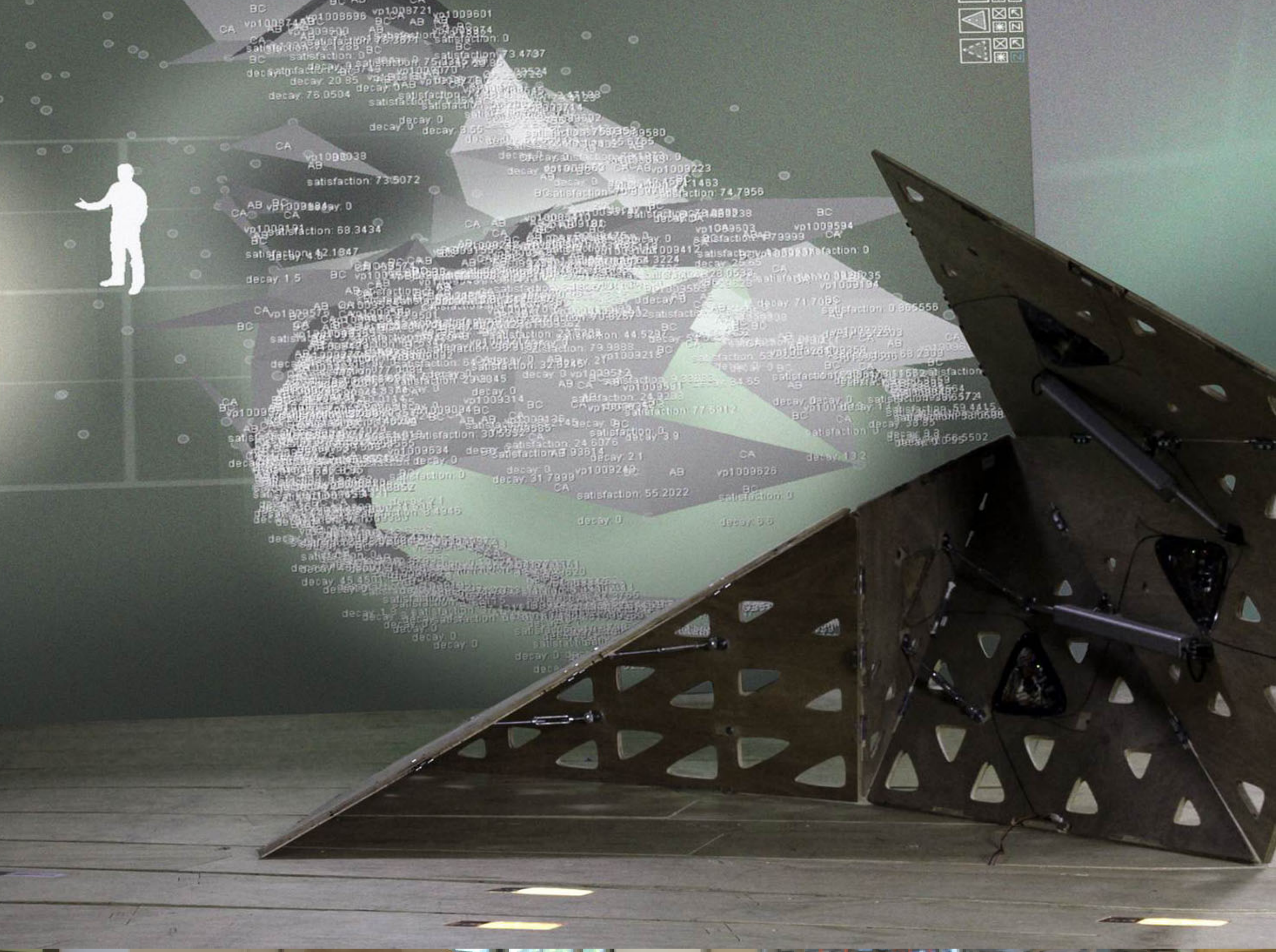
The design has been initiated by choosing a location for system deployment. In previous projects it has been observed that even if architectural systems were designed as generic systems, the deployment location would have very large potential influence on system development. Not only spatial or environmental constraints, but presence of specific visitors, social relationships between them and patterns of their activity would play the key role in project development or even be the key factor in project's success or failure in sustaining envisioned user activities and serving the role it had been designed for. In this respect users of the project need to be treated as integral parts of designed system.

The test site chosen for the first version of the deployment of the d|e|form system is the Mekelpark on TU Delft campus; the public park, which also serves as a main pedestrian communication artery of the university. Prior to the start of the project the park has undergone a radical transformation, following the design plans by the architecture firm Mecanoo. Description of the Mekelpark's intention on its designers' website says "Now, with the buildings located alongside the park, they illustrate their interdisciplinary connection. The park unifies and adds allure to the university while also providing a pleasant experience"[1]. The d|e|form project was conceptualised as an answer to the challenge of extending the functionality of public urban spaces in the context of TU Delft Mekelpark. Mekelpark is the main public space of the TU Delft campus, situated along the central axis of the campus. The Mekelpark is an attractive green space, yet, despite its recent renovation it is not free of problems. Although it is intended to work as a social hub for TU Delft students and employees, it mainly operates as only a pedestrian transition zone.

Daily author's observations conducted at the outset of the project showed that the designed "interdisciplinary connection" does not occur in any other than metaphorical sense. Even on a warm and sunny day, it was uncommon to find students sitting on the park grass, since the most popular hangout spot of the campus remained to be the green sloping roof of the main library. Many of the passers-by complained about the lack of facilities in the park. Their ideas for improving the situation ranged from inclusion of more benches accompanied by tables, canopies, small food stalls or even larger pubs, restaurants and clubs, or informal spots where half-formal study- or work-related meetings could be held. It had been concluded that many such needs could be met by provision of small scale urban furniture elements.

Consequently, the initial design concept was to design a system of urban furniture objects that can continuously adapt to their changing context, defined by all factors listed above. What's more, such urban furniture installations shall not only actively respond to emergent behavioural patterns and activities of their occupants, but also pro-actively affect how their premises are being used, actively encouraging park visitors to occupy the park and socialize among each other.

[1] 'Mecanoo Website ("mekel Park, Campus Tu Delft" Project)', accessed September 16, 2010, http://www.mecanoo.nl.


vp1009601
decay: 76.0504
satisfaction: 73.5072
satisfaction: 68.3434
satisfaction: 42.1847
decay: 1.5
satisfaction: 74.7956
vp1009594
satisfaction: 0.865556
satisfaction: 77.6917
decay: 31.7999
vp1009626
satisfaction: 55.2022

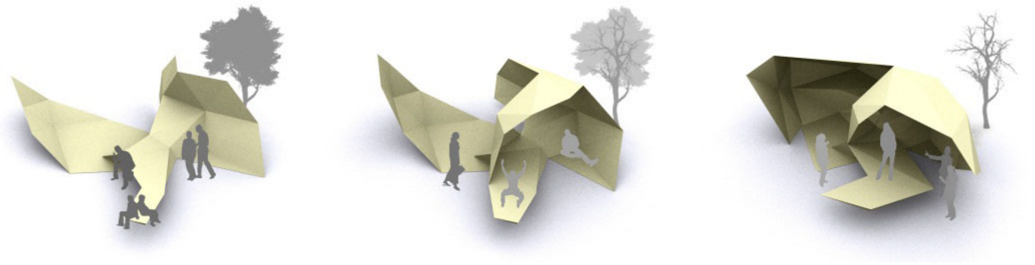


Fig.39. Sketched development of d|e|form system deployment over time

The design of a component was initially limited to its triangular geometry. Being the basic geometric rigid primitive, use of triangles allowed conceptual formation of a wide range of forms with little geometric constraints. A number of case study sketches were made in order to quickly evaluate possibilities of project's development over time and to assume a realistic scale for components and assemblies. As a result component size of triangle edges not exceeding 1,2m and not smaller than 0,3m was chosen. Such size guaranteed formation of all required spatial affordances while retaining manageable quantities at this design stage.

Consequently the system has been prototype virtually, while more detail was added to components. The detail included hinging connections between panels and a system of perforations decreasing weight and allowing opacity variation. Virtual panels were also provided with physics simulation, allowing basic real-time testing of the structural and mechanical dependencies between panels and the dynamic capacities for transformation of the installation as a whole.

Next the behaviours of panels were designed through virtual prototyping. The behaviours had two layers. The first layer governed dynamic adaptations of panels to sensed activities. Users were virtually simulated by author-controlled avatars. Components would sense presence of avatars on their front and back side and record duration of their stay. From that activity type was derived. Further, each component was given the ability to decrease or increase the relative angle towards its neighbours. Passing-by of users characterised by relatively short presence would trigger components to move away giving users more space. Longer presence of users would trigger panels to move closer forming a more intimate surrounding and providing places to sit down or lean against. Additional virtual agents were introduced as "space particles" which would communicate with surrounding them panels and record long term presence of users, creating a dynamic activity map. Initial behaviours of panels were not including communication between them. After iterative development and testing cycle, local communication between panels had been enabled, allowing them to locally synchronise their actions, avoid collisions and erratic behaviour. This allowed to change the initially chaotic behaviour of the structures as wholes to more fluent and integrated transformation.

The second layer of component behaviour was aimed at creation of new panels and removal of old and redundant ones throughout system's operation. For this purpose each component would be equipped with internal evaluation mechanism. The usability variable would increase every time the component is used and would be passed on to neighbouring components and beyond at decreasing value. At the same time if not used, component usability would slowly decrease. If the value would drop beyond a centrally established threshold the component would mark itself for removal. If the value would grow beyond a certain threshold, the component would signal that a new component should be added to one of its empty edges if available, or otherwise send a usability variable boost to its immediate neighbours. The algorithm required iterative adjustment and extensive virtual testing before the desired balanced growth and dynamic adaptation were achieved.

Img. 10. Left: Tomasz Jaskiewicz, d|e|form project, testing interactive behaviour and user-driven assembly process.

The next step was design of the physical panels. Plywood was chosen due to its affordable price and ease of cnc fabrication. A test batch of panels was fabricated based on one of the virtually prototyped assemblies. Consequently with help of a group of students the installation was assembled and disassembled. In the process the weight and strength of panels proved inadequate for the size of the assembly. Despite ealier simulation, the furniture hinges and drill mounts in plywood were not strong enough to withstand some of the stresses coming from the structure when being manually reconfigured. As a result a new design was made and the prototype has been downscaled for further testing of autonomous operation.

Seeduino mega boards were used as embedded controllers. Electric actuators were introduced to control the angle between panels. Components were communicating between one another through three hardware UART connections and debugging and central control were done using a wireless ZigBee network. In both cases a high level protocol for communication among components and the remote controller had to be established. For the purpose of the installation a simple convention has been established.

The assembly of six components was tested in conjunction with their virtual counterparts. Clearly, the material aspect of the installation required improvement, as not enough rigidity of the structure was there to support sitting or standing on it comfortably. The re-fabrication flow worked technologically, as new panels were being virtually generated and fabrication files automatically prepared, while unused panels were marked for deletion. However, testing the application has shown little valuable input in respect to usability of the system and user interaction in the actual context. Interviewed visitors admitted not knowing what the intention of the structure was and not understanding its affordance as furniture or transformable multi-purpose object without explicit introduction. An improved version of the prototype would be required to deploy and test it in the actual spatial and societal context. User-interaction mechanisms were lacking to solve the problems of maintenance and supervision. Ideally users themselves should be given incentives to perform those operations.

c) Evaluation

Rather than being conclusive, d|e|form project has raised the whole range of methodological questions to be answered. It has shown the feasibility of parallel design, virtual and physical prototyping by a single designer. Yet it has also demonstrated the wide number of open questions that follow. The transition from a working prototype to the working installation has proven to be a complex endeavour. Technologically working system of components still requires a system of inhabitants to use it and perpetuate its development. The involvement of those inhabitants is therefore critical from the early stages of the design and prototyping process. Gradual diminishing of the role of the designers and experts combined with increasing role of inhabitants in architectural system's development and maintenance appears as the most desirable scenario.

4.5. Discussion

In all projects presented in this section, processes of aggregation of components into architectural interventions have been investigated. Initially these processes were conceptual; consequently they were enhanced with virtual and actual aggregation of system elements. The traced processes have demonstrated the difference in approach from this in traditional architecture. In systems where emergent properties and bottom up development are desired, the actual deployment (virtual or physical) was essential in order to estimate system's performance and advance the design and ensure its balanced further development. It has also been explicitly demonstrated that in the initial project design and development such balance is difficult to achieve, with the difficulty level increasing with the number of components. Centralised control of every component becomes impossible in larger systems, distributed systems are open to unexpected bifurcation and can easily fall into a positive feedback loops causing system instability. Because of this self-regulating mechanisms and negative feedback loops need to be implemented and tested in early design stages.

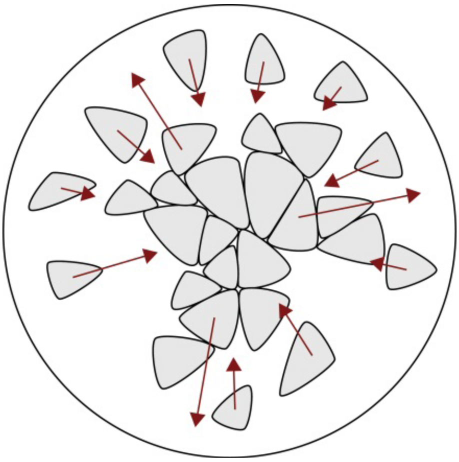


Fig.40. Ideogram: continuous, iterative assembly, adaptation and disassembly of components

All traced projects envision highly adaptive architectural systems. The combination of component dynamic flexibility and gradual addition and removal of components over time presents a solution where both short-term adaptation and long term growth and eventually evolution of the designed system are possible. Yet only further tests and advancement of prototypes and deployed systems can provide auxiliary evidence. The new problem, which has presented itself in the d|e|form project, has been the question of protocol. For the purpose of d|e|form an ad-hoc solution could be used. However, further development of the system would require an approach ensuring scalability of communication between components as new features are added to the system, or as several systems merge.

However, the main difficulty in respect to further design of distributed adaptive architectural systems lies at unpredictability of user behaviour in these systems, which in itself can easily fall into a positive feedback loop. Ways need to be found to include active users in the designed and prototyped systems from the earliest design stages.

Conclusions summary:

- Centralised control and linear development scenarios are not possible for large adaptive systems.
- Flexibility and interchangeability of components can complement each other for combined speed of adaptation and high degree of adaptability.
- Virtual simulation can be used to locally anticipate system transformation and guide the system into a stable state by locally inducing negative feedback loops.
- Shared protocols are needed for connecting components in respect to both the physical connection and information exchange.

5. Participants

Challenges:

- Participants need to be inherently accounted for as part of designed systems, initiating and steering the development of architecture over time.
- Ways for involving or/and simulating participants need to be investigated.

The main interactions in projects traced so far in this chapter have been occurring among designers and the designed system agents, namely spaces and building components. The designers clearly operated as a specific kind of system agents, also as participants within the systems. However, conventionally the role of designers is not to participate in systems, but to form them for participation of others, its inhabitants. As concluded in the previous section, it is essential for architectural systems of potentially changing, adaptive, functionality to take into account the actions of inhabitants already in the early stages of system formation.

Most commonly expert designers (including architects), clients and inhabitants are different parties in the project. However exceptions occur frequently. The current trends are, for inhabitants - to organise themselves into bottom-up project developers and for architects - to allow more participation of inhabitants in the design processes. In this way inhabitants become not only clients, but begin to operate as designers with unique expertise.

Design of a building as a flexible system allows inhabitants to assist or even hypothetically replace the expert designers in the process of formation of architectural systems they inhabit. Assembling and consequent modifications and “tweaking” of an architectural system is an activity in which inhabitants and/or clients could participate. As much as creation of agents and their behaviour is a task requiring specialized knowledge, typical agents and templates could be introduced to create an extensive library of agents out of which specific building systems could be assembled. Such was the case in earlier described project Manhal Oasis, where representative of clients were encouraged to modify the distribution of functions in the project and which resulted in significant adjustments in the final design.

Examples of involvement of inhabitants in architectural design are widespread. When needing to modify the interior of their house, inhabitants often design or re-design that interior themselves. Firms such as IKEA provide them with software tools that permit virtual distribution of furniture agents without the need for expert skills, followed by automated ordering of house interior elements fit to the specified arrangement. When new land use plans are being designed, inhabitant representatives are often being invited to design sessions. In most western countries inhabitants also have the legal right to propose changes to a land use plans at the end of their formulation, before they become legally binding.

However, in many cases the direct involvement of all current and future inhabitants of the project site may not be possible during the design phase. Also, inhabitant input is often limited. Can provide intentions of inhabitant during the design process, but does not directly guarantee that in different conditions inhabitants would not provide other inputs. Their behaviour can change over time through interactions with other inhabitants, the developing habitat itself and external environment. The Paracity project was developed by the author as a large-scale architectural system. The aim of Paracity was to explore inclusion of virtual participant agents to complexly formed architectural systems.

5.1. Paracity

a) Outset

The Paracity[1] was a project conducted for author's MSc graduation thesis in 2005. It was aimed at designing a feasible plan for the expansion of the city centre of Gdynia (Poland) to the former industrial area of Miedzytorze. The centre of Gdynia is a vibrant area, where shops, services, offices, housing and public leisure zones interweave. The shops are mostly clustered along the main street Swietojanska. During the decade preceding the project, Gdynia has faced a dynamic development of its centre. Offices and shops took over former housing buildings. Public spaces were intensively redeveloped. Much of that development was not planned top-down. New functions self-organised in the city. The premise for the project was to develop a thorough understanding of these mechanisms and to plan the new fragment of the city so that it would accommodate for the city's emergent and unpredictable formation and "sedimentation" of urban programme in its fabric.

b) Process

From the start the project was developed through seven "views" on the designed system. The views operated as containers for agents of specified type, while allowing these agents to communicate across the view boundary.

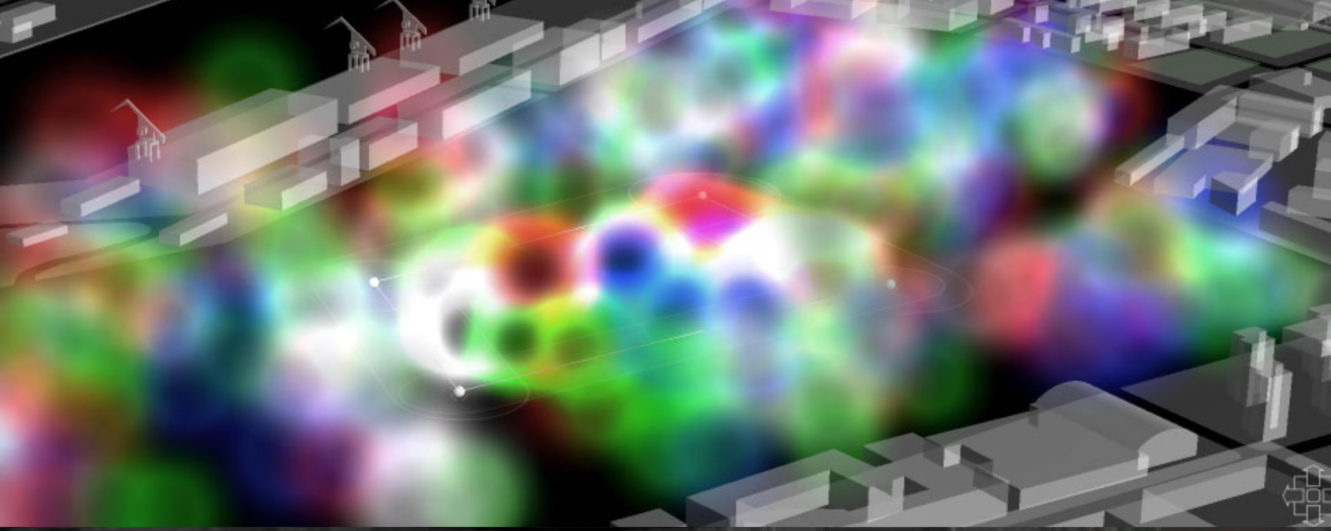
The "context" view dealt with agents external to the site, which were considered to have a significant influence on the redevelopment of Miedzytorze. The "networks" was a view bringing together communication infrastructure agents, mainly roads and pathways. The "program" view hosted all functional program agents. These views allowed to distribute the functional program on the site in a way similar to ONL projects discussed in point 1.2. The Addition was the inclusion of the external agents and the network agents which functioned not only as attractors in the system, but also allowed to build relations between the elements of the site and specific agents. As the intention of the project was less focused on creation of architecturally expressive forms, but more to devise the inner workings of the urban system, powerlines were not used in this project, although the agents used to define the infrastructural networks had many similarities with powerlines.

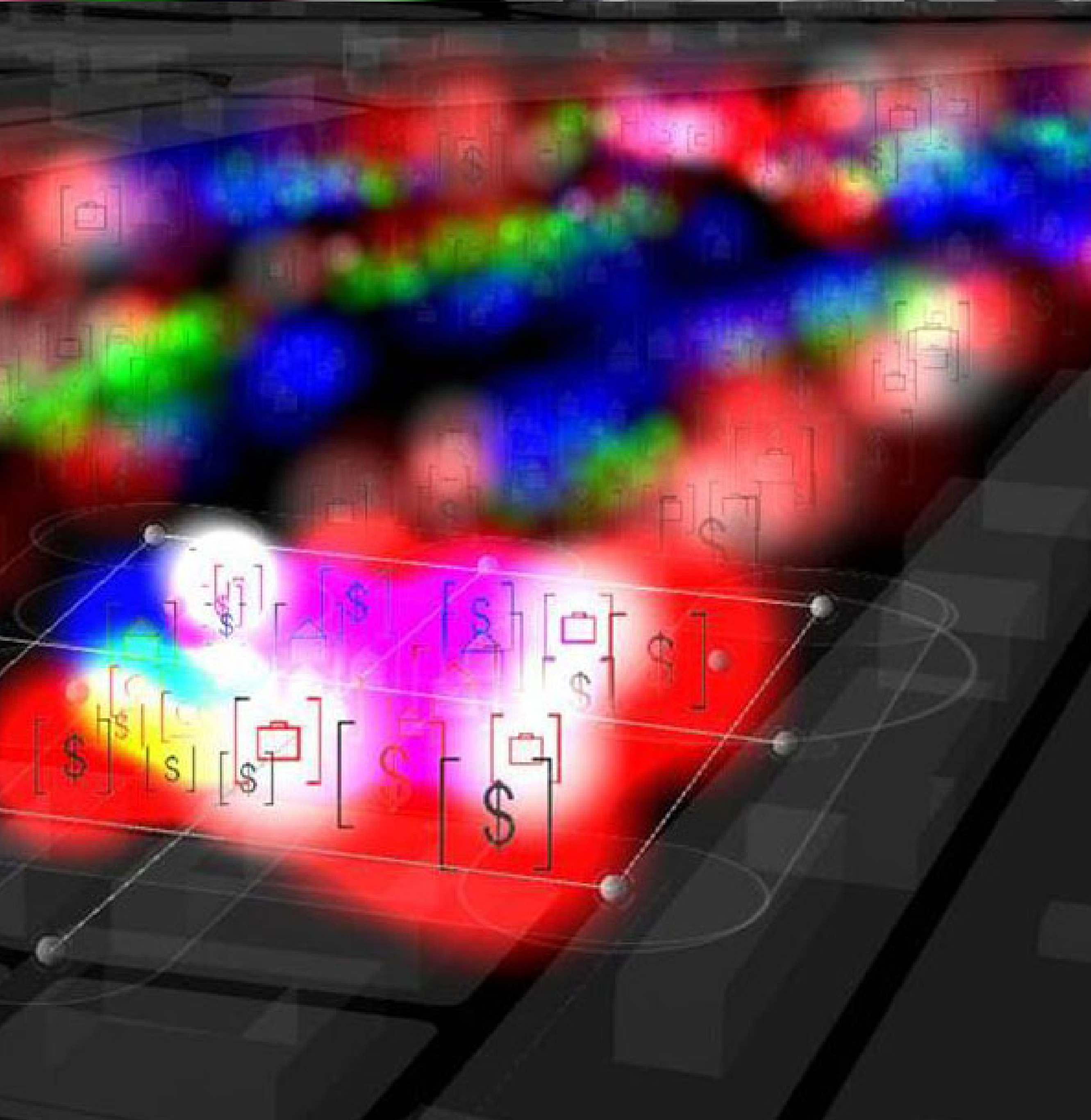
Through the "networks", "context" and "program" views the system had been developed by the project designer (the author in this case) through addition, organisation and local interactions with individual agents and control of the generation of new agents in the process. However, such approach by itself would have risked neglecting the urban dynamics coming from the city inhabitants and its role in the formation of such a large and vibrant city area.

For this reason an "inhabitants" view was added. Clearly, due to project constraints, it would not have been possible to involve hundreds or thousands of real city inhabitants in the formation of the plan. Therefore, as an alternative, the simulated user agents populated the "inhabitant" view. The behaviour of scripted virtual agents was not aimed to exactly copy the behaviour of humans. It was created in a way that reduced human behaviour to only the most basic factors, which, following earlier studies, governed the likelihood of emergence of specific types of the functional program in given areas. Inhabitant agents were thus continuously moving through road and pathway infrastructure of the city while trying to reach goals picked by a statistically weighted random function.

The movement of inhabitant agents occurred continuously. As the city system developed, inhabitant agents would instantly begin to operate in the changed environment. Based on the statistics on these movements, the next "program sedimentation" view would automatically generate urban functions based on the statistical threshold, and remove other functions

[1] Tomasz Jaskiewicz, 'Paracity – A Digital Urban Design Process' (Delft, the Netherlands: Episode Publishers, 2006).





which were statistically expected to discontinue in the new conditions. To give an example, shops would flourish in area of high inhabitant agents activity, while housing would partly decline there, increasing in areas with considerably less “inhabitant agent” traffic.

At any time the state of the system could be displayed through “statistical” and “masterplan” views. Agents in these views were purely representative, integrating information from other agents and providing an overview of the current state of the system.

The project was developed by the author in three rounds. Each round started with a different set of initial conditions, and took a different path of development.

The first path was initiated by extension of the existing main shopping artery Swietojanska towards across the entire Miedzytorze area and gradual addition of side streets. The inhabitant agents densely populated the extended Swietojanska and clustered around its end. The author extended the network of streets in that area, as in his intuition it was a highly attractive location with spectacular vides over the port. Rapidly, commercial functions begun to emerge in the area and offices started to be generated in the perpendicular streets. Over time leisure programs and housing started also to appear. Gradually, the program became very dense and street network was extended further towards the west. These streets provided new connections towards the area of the city’s central train and bus station. This generated movement of inhabitant agents along two paths, which consequently became populated with shops. Housing areas began to blossom on the edges of the site. Numerous leisure areas appeared in between them and in proximity of the commercial zone. In several places chaotic self-perpetuating feedback loops started to form, where different functions would perpetually replace one another, causing a recurrent, non-stabilising fluctuation of the system. As the lead designer author had introduced a number of attractor agents that consolidated the fluctuating developments. These attractors could be compared to local land use rules that can be introduced in city areas where development becomes chaotic. Ultimately the plan has stabilised, the routes of inhabitant agents maintained on once established paths, which allowed stabilising the functional program of the city.

The second development path took a different route. Instead of extending just one street first, the entire street network of the city was being extended gradually towards north. The distribution was more balanced. New developments attracted few inhabitant agents, gradually filling up with offices, housing and leisure. Some of the new functions emerged into clusters, attracting more agents and eventually leading to generation of shops. As the network was expanded further, new connections between city parts were formed. The small commercial developments along these connections increased in size and generated more commercial and office functions. Eventually a gradation of density filled the streets from south to north. Office-commercial clusters stabilised.

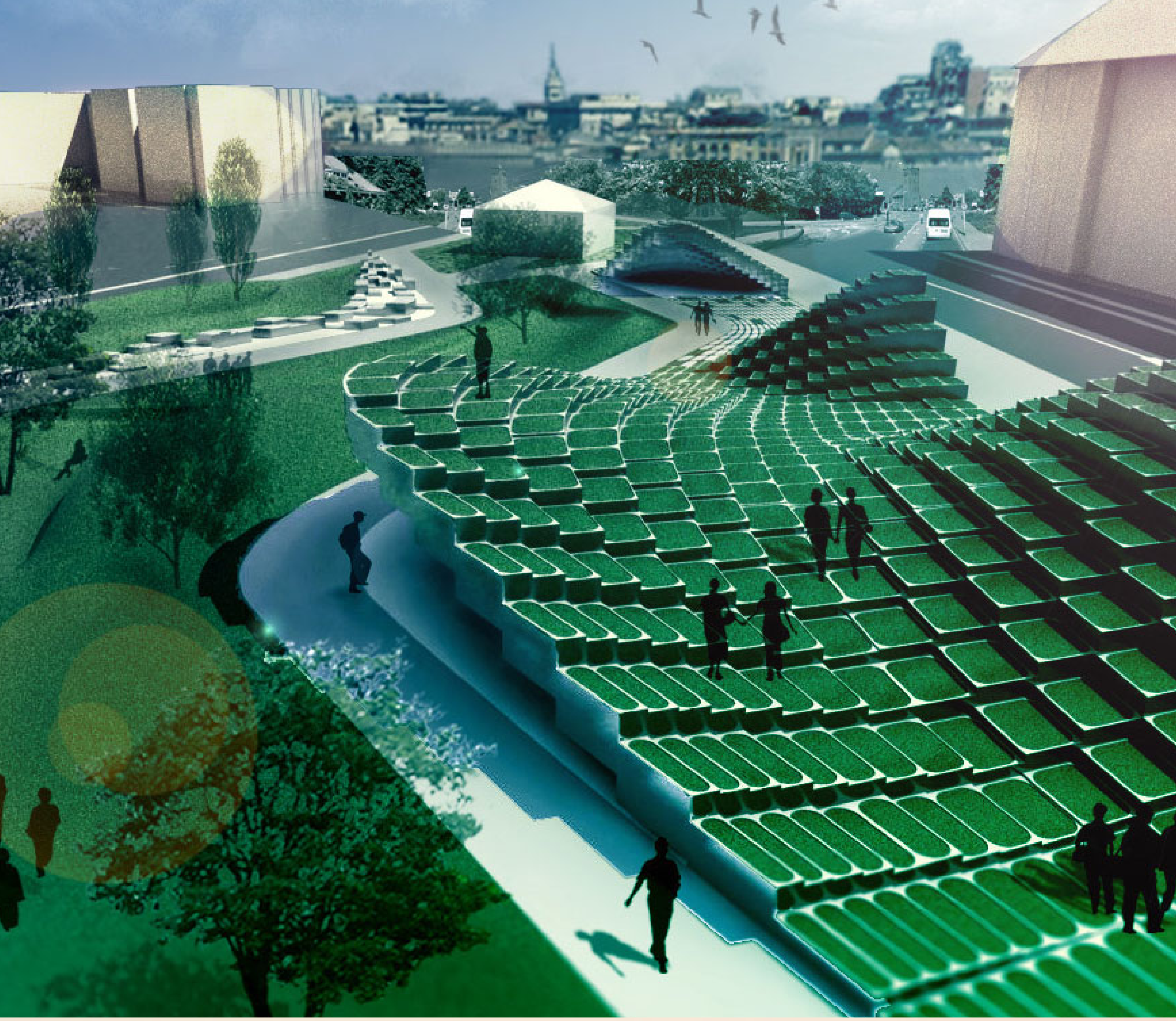
In the third path of project development a similar initial condition was assumed as in second, however the extension only continued to the boundary of the at the time functioning rail tracks, assuming the scenario of commercial revival of the port. In that case the commercial densification of the area was stopped, and streets and blocks near the border never became as active as in the previous scenario.

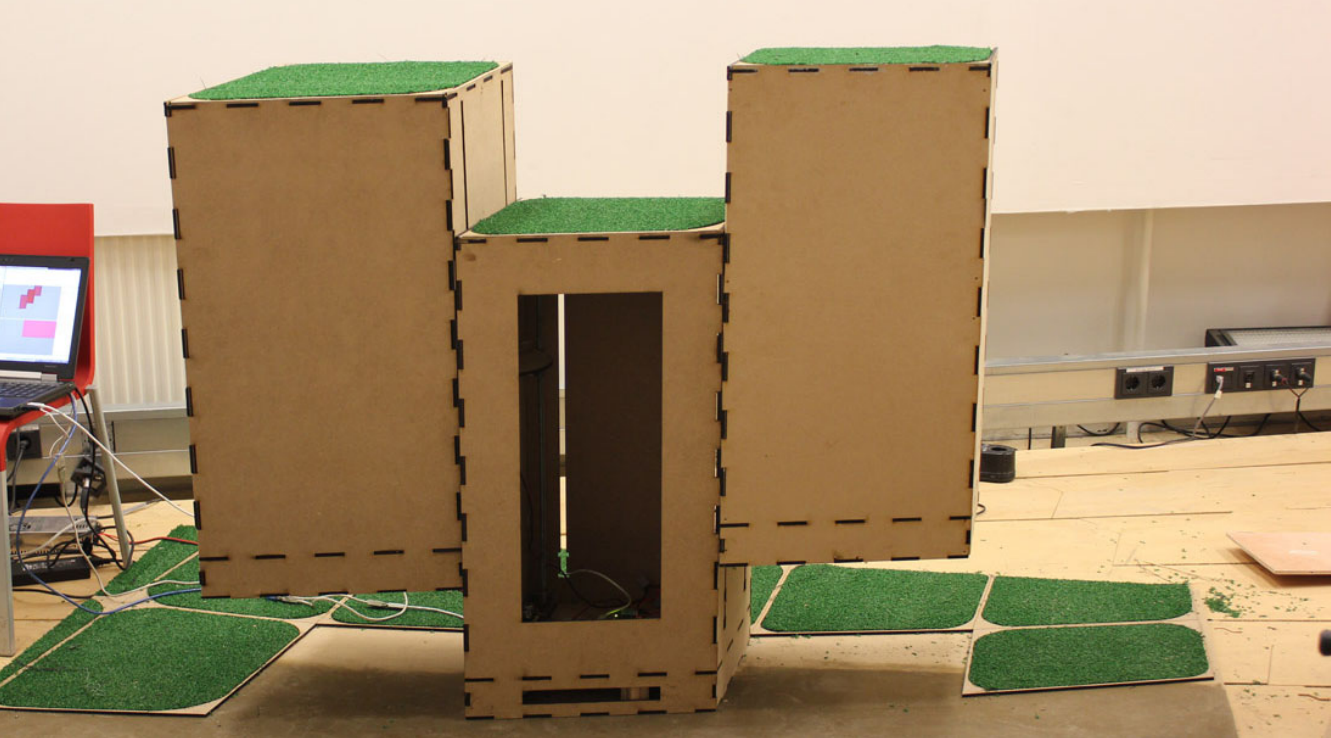
c) Evaluation

The Paracity project has delivered three scenarios of Gdynia city centre development that were well received by evaluating them experts specialising in Gdynia’s urban redevelopment[1]. This proved that the approach taken can be directly and efficiently applied to current urban planning problems. However, the true significance of the Paracity project lays in the approach to the urban design as a continuously development of a non-deterministic system of a city.

[1] Project was positively reviewed by dr. Piotr Lorenz and prof.dr.Mieczyslaw Kochanowski

Img. 11. Left: Paracity project





Qualities of such system are emergent, they cannot be controlled, but can be steered once the local interactions and the ways of their aggregations are understood. The Paracity project provided a valid method for dealing with such approach to urban systems.

The resulting plans provided possible scenarios in which the city centre could be gradually developed, ensuring that this development would account for dynamic factors in the city fabric and local economy. Clearly, the inhabitant activity simulation correlated with function emergence simulation could only had given an indication how real world urban processes would unfold in the city, but earlier studies have shown it does offer high accuracy in analysing existing city development scenarios.

The strength of presented projects lies in the ability to study the development of the city as a process with many bottom-up factors, providing novel ways of moulding this process as it unfolds. In reality it would have been possible, and significantly more reliable, to replace inhabitant simulation with feedback from the actual users and environment. In such case the simulated agents could be replaced by streams of data about site usage coming in real time from the project site.

In the context of presented case study research, the Paracity project demonstrates the development of an architectural project with taking into account the dynamic and unpredictable nature of the activity patterns of its inhabitants. It shows that in complex spatial organisations, the ways in which these activities unfold in space can be difficult to predict, while at the same time these activities are the determining factor for the success of planned development.

Traced development scenarios demonstrate how the city development process changes when the feedback from users, simulated or potentially actual, is provided continuously throughout the plan development process. It is shown, how the complex process of the city can locally adapt to such activities as they unfold in order to deliver a robust urban system.

5.2. VHpark

a) Outset

The VHpark project has been initiated to propose a number of possible alternatives for redeveloping the de Vries van Heyst-plantsoen, a small park surrounded by roads and located between TU Delft's faculty of Architecture, the Delft Science Centre building and a student housing complex. By Sepember 2011 when the project has been initiated the park has had been unused infrequently, despite its prominent location on the border between TU Delft campus and Delft centre. As identified in initial analysis and questionnaire numerous visitors to the area indicated that it would have been a perfect location to boost the frequently complained upon social life of the area, where students of TU delft could mix with creative start-ups employees and local inhabitants.

b) Process

The project has been given as a study assignment to a group of 31 Hyperbody MSc1 students divided into 11 teams, further referred to as "atoms". Each such atom would be initially responsible for identifying the architectural opportunities and problems found in and around the park. Such identified design challenges had to be supported by "validators", being either external experts in relevant domains, or actual inhabitants of the area providing their opinions through interviews or surveys. In synchrony with defining the design challenge, students were developing initial design concepts, that in turn would steer their preliminary research and help in communicating redevelopment possibilities to the validators. It was also

Img. 12. Left: Flexscape project, design and prototype

required of the students to document their work by continuously updating the online wiki pages of their atom, where the structure of the entire wiki website was provided to them at the start of the course, but atoms were entirely responsible for respective content.

The wiki platform has facilitated tracing each atom's work, since every update to the website is recorded in its history. It could be clearly observed that there was initial resistance to the unknown working procedure and confusion among students. Nevertheless, it has been quickly picked up by most of the atoms. In the first weeks of project development numerous interviews were made with passers-by on the actual project location, either transcribed or recorded on video. Other atoms used surveys to collect feedback from passers-by. Common critique was that only limited number of users could be addressed in this way, making it difficult to develop a greater understanding of the patterns of current behaviours of park visitors and of users of leisure and social places in general. In response to this ciritique the atom of Niklas Ruprechter, Jacob Lam Chak, Lieke Kraan and Liviu Teodorescu has recorded an aerial video of a 12 hour cycle of park use, from which patterns of visitors passing through the site were extracted. The atom of André Dessens, Frederich Steenkamp and Michael Zhang performed a similar study but with the focus on more general social patterns occurring in public spaces in conjunction to spatial affordances. The video was recorded in the vibrant and open Oostserre area of the faculty of architecture, where its makers would continuously change the organisation of furniture elements and observe how it changes the social behaviours of continuously streaming in groups of visitors.

Only few groups connected to experts. This was explained by no clear need for the expertise or lack of entry point for makning connections with such experts as e.g. sociologists who could have helped in initial research. Clearly, lack of earlier experience also contributed to the difficulty of students in establishing links with people beyond the internal system of the faculty being their comfort zone. Cultural and language differences resulting from many of the students being non-Dutch did not seem to provide serious obstruction. Mixed groups of Dutch and non-Dutch students allowed to minimize such drawbacks.

The design concepts addressed different ways in which architecture can be used to bring people to the park. Ideas ranged from providing secluded spaces for individuals or small groups to providing spatial affordances for big entertainment events and exhibitions. Temporality of projects also greatly varied. Some proposed permanent structures while others envisioned entire park to be rapidly transformable over time. Also means of achieving desired spatial effects varied. They included using passive acoustic properties of space as an enabler, organising social behaviour around making fires and playing collaborative games, various forms of kinetic adaptation or complete crowdsourcing of the project through an online platform and a set of tools and interfaces.

In the second phase of the semester, atoms were encouraged to connect to each other and form hybrid projects combining qualities and challenges that atoms individually earlier identified. This however occurred only in one case, despite the benefit of less workload per person in connected atoms. As a reason student mostly provided the argument that by the end of the first phase their design concepts were too distinct and incompatible with each other. The second phase ended by development of complete architectural projects. After tracing the development of the projects with help of the wiki it could be observed, that despite seeming autonomy of projects, there were some mutual influences. One of such similarities was the "sectioned" from design approach, also related to the popular 3d modelling technique that students learned and exchanged in accompanying the design studio architectural studies course. On the other hand the relationship of developed projects to initially identified validators and their feedback remained superficial at most. Despite being frequently steered to do so, students have not actively involved external participants in the designed projects. As a result projects provided flexible and adaptive solutions aimed

Img. 13. Left: Spacebook project, design and prototype at various kinds of social integration of park visitors. However, the designed solutions were based on subjective and often ungrounded opinions of students, not verified by in-depth study or further questionnaires.

In the third phase of the design assignment students were guided to validate and further develop their projects using full- or large-scale prototypes of the most representative aspects of projects. As a result each project has been prototyped as a section or cluster of components. While prototyping, designs were being adjusted, based on material performance evaluation and refinement of actuation mechanisms or interfaces. Nevertheless, little feedback from actual validators was provided in the process. Following discussions with students it occurred that there was a mental barrier to bring "external guests" to give feedback upon unfinished prototype and/or such activity was regarded by students as too time consuming to be afforded in the short time span of the project.

c) Evaluation

All projects have been highly innovative in their ideas and reasonably advanced in regard to their execution. Nevertheless, none of the projects could be considered ready for realisation.

Each project focused on a selected aspect of the complex challenges offered by the project location. In this projects were guided to approach the challenges comprehensively, among others proposing business models and analysing the social and cultural impact of designed interventions. Ultimately the projects have addressed the complex issues involved in the proposed interventions to a much higher extent than it would be the case in typical architectural projects. However, due to the unconventional aspect of interactive architecture, would not have sufficed to deliver convincing evidence of commercial and societal viability of designed projects.

What's more, globally, it remained debatable whether organising a team of designers to propose alternatives for the redesign of the VHpark was the right strategy. On one hand designed projects have shown a variety of possibilities for the park. On the other none of the projects provided a coherent and complete vision for complex integration of a variety of aspects and possibilities.

Most importnalty, however, the projects have demonstrated one of the significant challenges of participatory design. Inhabitants and experts have proven to be difficult to be engaged in the design process and prototype evaluation and feedback. The projects shown that extensive persistence is required from designers to gather such feedback and new ways need to be found in order to structurally involve inhabitants in designed for them spaces, especially considering that in realised projects such continued engagement is critical for the success of an interactive architectural building.

5.3. Discussion

Designing city-scale projects based on complexity theories in conjunction with virtual reality environments is a growing research area[1]. Two traced projects have principally demonstrated two aspects in which users can be accounted for in complex architectural projects. Paracity project has integrally used simulation to anticipate complex distribution of pedestrians across urban space in reciprocal relation to emergence of the functional programme around the designed public spaces of the city. The VHpark project, consisting of 11 alternative design proposals, has attempted to involve its prospective users directly and gather their feedback across different design stages.

[1] J. Portugali, *Complex Artificial Environments: Simulation, Cognition and VR in the Study and Planning of Cities*, 1st ed. (Springer, 2006).

As shown, each approach has advantages and disadvantages. Multi-agent user behaviour simulations, once developed, allow prediction of complex feedback loops occurring between many users and many project components. Such predictions are impossible using standard methods. However, the simulations are highly limited in being able to address only few aspects in user behaviour, e.g. mobility in case of Paracity. Real human beings, often act differently than simulated agents, since their behaviour is a result of complex history of individual experiences that include the cultural context and one's individual personal traits. In this, human beings may on observation not act as understandably rational beings.

On the other hand, direct feedback from users guarantees that all aspects of that person's experience will be reflected in that feedback. However, unless the project can be fully realised, participation of multiple users and all aspects of the project cannot be validated.

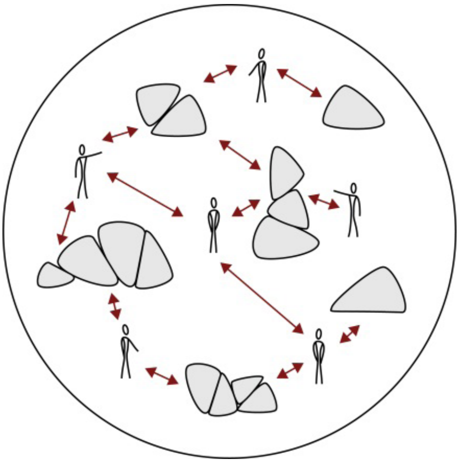


Fig.41. Ideogram: dynamic network of interactions between participants and building assemblies drive development of complex adaptive interactive architecture.

What's more, it has been shown that acquiring feedback from actual users is a difficult challenge, especially considering the need for reclusiveness of one's feedback. Numerous feedback gathering methods can be encountered e.g. in user-driven product design. These methods however become additionally difficult to use in architectural setting due to the high intricacy of projects and individuality of associated experiences.

In conclusion, new ways need to be found for acquiring feedback from users early on in the design process, continuously enhancing it into the prototyping and project deployment. At the same time, especially in the early phase of project's design, direct feedback needs to be supplemented by simulation, where mutual connection between simulation and user feedback need to be established.

An additional observation in both Paracity and VHpark projects relates to the locality of human-architecture interactions. In case of Paracity, that locality was the key outcome of the simulation, leading to emergence of functional programme of specific kind based on location of simulated users in the area and duration of that location. On the other hand, all projects of VHpark were highly location specific. Feedback from users was given for the concrete location of the VHpark, not for a generic park which could be placed anywhere. As investigated by student, the specific mix of users in the VHpark area has led to the unique project challenges, wich would differ if applied to another park area. For that reason, additional attention needs to be given to the location model[1], where interactions between users and architectural components are associated with a specific place.

Conclusions summary:

- Feedback gathering models need to be found from actual future users of the designed project.
- Multi-agent simulations can provide insights into aspects of user participation, but are highly unreliable and only indicative.
- Cultural conventions and mental models of static and "permanent" architecture need to be overcome in order to receive unbiased feedback.
- Techniques for providing participants with specific information on designed project and focused feedback formulation need to be found.
- Location model for components is needed.

6. Assemblies of architectural spaces

Challenges:

- Lack of intermediate structure to formulated dependencies between people and components.
- Processes investigated to point don't include the concept of architectural spaces and their organisation.

The following three projects have been developed at the architectural firm ONL[Oosterhuis_Lénàrd]. The author has participated in these projects and performed empirical methodological research on multi-agent formation of spatial organisation in architectural systems. All three projects employ a design method, which is an extension of the method presented in point 1.1. Experimental software prototypes created by the author have been used to facilitate investigated design processes. These prototypes will be discussed in detail in chapter VI.

6.1. ONL Salzburg National Park Centre project

a) Outset

The Salzburg National Park Centre (SNPC) is a tendered competition entry project by Oosterhuis_Lénárd[ONL]. The project was executed in the early 2005. The project is an answer to problems defined in the typical briefing procedure in which architectural competitions are organised. The functional program and site were given as fixed and the role of architects was to devise a sound and aesthetically attractive spatial organisation and materialisation plans of the building.

b) Process

The design process became initiated through an interaction of two agents; the competition tender can be seen as one of them, including all the rules of the to-be design. The office principal, prof. Kas Oosterhuis can be seen as the other agent, who following the reception of the tender became the lead designer and coordinator of the project. Following his initial

[1] Malcolm McCullough, *Digital Ground: Architecture, Pervasive Computing, and Environmental Knowing*, New Ed (The MIT Press, 2005).

interaction, the team of ONL employees formed to work on the project and the author was invited to provide the multi-agent environment and related to it design methods for the development of the project. The author has also been given the opportunity to monitor the process.

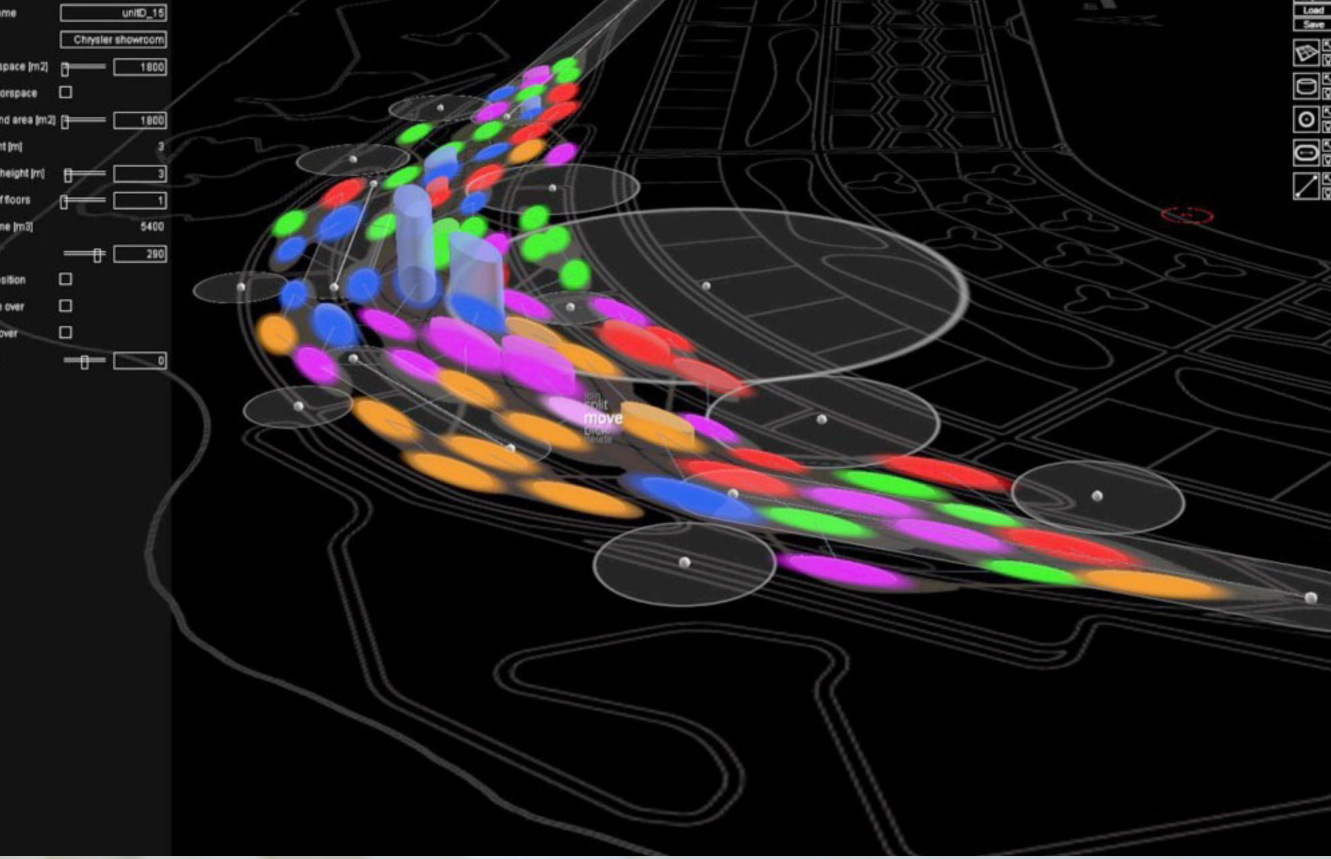
The design has been initiated by creation of one virtual agent. This primal agent represented the assembly of all functional spaces of the designed building. It encompassed all constraints and functional program rules coming from the brief. The initial agent has been subsequently subdivided into specific functional clusters and eventually into individual spaces. This led to a set of over hundred distinct building spaces, each represented by an agent with floorspace and position being its main parameters. Each agent would be locally aware of its own properties. It would also know to which other spaces it should be directly and indirectly connected to and it would have an ability to query these spaces for their particular parameters. The task of a designer (at that stage Pim Marsman was operating the virtual multi-agent environment with frequent feedback from prof. Kas Oosterhuis) was not to define the elements of the program and fix them in place. It was to keep adjusting their parameters and relations and interact with the program model to achieve a spatial distribution of the program.

Behaviour programmed (by the author) into individual cells of the functional program was simple in its principles and became adjusted throughout the design process. Each agent had to avoid collision with other agents. If that was to happen, the cell would be forced to move away from the collision point. Connections between particular cells were formed in a top-down manner. If two cells were connected, they would try to stay closer to each other than the specified maximum and further away than the specified minimum distance. Additionally, spatial attractors; points and lines, were introduced to the system in order to repel or attract all, or just a selection, of cells to specific areas in the plan. It is through these attractors, that designers could impose additional, subjectively defined constraints on the system.

From individual, simple local actions and interactions of functional program agents, a complex, holistic system came to being. Even though individual rules and behaviours were straight-forward, the system as a whole exhibited a high degree of adaptability. Upon alteration of individual parameters, the entire system would reconfigure itself, in effect providing unexpected spatial organisations of functions within its boundary. This organisation, however, would always remain a logical outcome of predefined rules and boundary constraints, thus was guaranteed to be a well-performing one.

While the distribution of the functional program coming from its decentralized behaviour was entirely emergent and unpredictable, attractors and explicit relations between selected cells were used to bring the whole system to a stable state, by adding deterministic qualities and imposing constraints on it, in order to allow its operation within chosen development scenarios and an overall design vision. Recalculation of global parameters such as the total floorspace or volume of all cells together allowed changing these values locally while maintaining strict control of the overall project vision and its feasibility.

The interaction between the team of designers and the system of virtual spatial agents allowed reaching a robust organisation of the functional program. This process unfolded alongside the design of a styled, visionary form of the building, which had to contain the spatial arrangement of the program. A number of “powerlines” were drawn in three dimensions by prof. Kas Oosterhuis and office partner Ilona Lénárd around the “swarm” of virtual building functions. From that point on, the program distribution has stabilised. The detailing process continued linearly by gradually adding more detail to the project and in a hierarchical manner maintaining parametric relations between added building elements and functional program agents. In this process the powerlines were used as guiding trajectories for the building form. A parametric model of building components including the structure, internal walls, cladding panels, floor slabs, doors and windows was constructed under the lead of Gijs Joosen and



ome unitD_15

Chrysler showroom

space [m2] 1800

orspace □

nd area [m2] 1800

nt [m] 3

height [m] 3

f floors 1

ne [m3] 5400

290

sition □

e over □

over □

0

Sander Boer. Parametric relations created in that model were one-directional, meaning that a change in program distribution could be to some extent reflected in adjustment of the detailing, but changes in the detailing were not reflected back in the arrangement and parameters of the functional program agents.

c) Evaluation

The design of the SNPC project has been a fast two-step track. The first step has only involved the programmatic distribution and self-organisation of building programme. It has been the interplay of internal forces between program cells and forces provided by spatial constraints imposed by designers. The second phase provided the architectural form made of a system of specialised components required to physically materialise the designed spaces. Dynamic nature of the program system and parametrically constructed form allowed certain degree of adjustment of the project after the two phases were complete. However, that adjustment has been constrained by the integrity of designed form.

The first phase allowed high adaptability of the created network of program units. Shall properties of some components be changed, new relations added, some components removed or new ones introduced, the system would instantly adapt to that change. That degree of adaptability has been lost after the introduction of the building envelope, which was able to accommodate some parameter changes, but not topological reconfigurations of the program swarm.

6.2. ONL Speed and Friction Automotive Complex project

a) Outset

The Speed and Friction Automotive Complex project by Oosterhuis_Lénárd[ONL] followed a tendered assignment to create a masterplan for the site located on the outskirts of Abu Dhabi, covering more than 6 km^2. The project was executed in the end of the year 2005. This large site came with a large program of demands. 800.000m^2 of covered showrooms, car shops, car-themed leisure and experience destinations, restaurants, a hotel, a conference centre and two gas stations. All was accompanied by over 500.000 m^2 of covered parking spaces and many more outdoor facilities such as a formula 1 and 4x4 race tracks. All of this had to be organized into one architectural plan proposal, while retaining a broad margin of flexibility as to exact location and floorspaces of planned functions, required by the commissioning developers. Ultimately, the project not only had to work as a well-organized system of car-related services and entertainment areas, it also had to become the landmark of a global destination hub for all car enthusiasts with thoroughly consistent and appealing aesthetics.

b) Process

The design process of the Speed and Friction Automotive Complex proceeded similarly to the SNCP project. Its development followed the same general design methodology and was conducted in the same design software environment. However, the scale of the project was approximately 800 times greater. The functional program and corresponding floorspaces were indicated in the brief, but not precisely defined. Prof. Oosterhuis, assuming again the role of the lead designer, took the first step in the interaction with the project system – consisting initially of one “global” agent at that point. Based on estimate calculations he attributed the global floorspace to the initial agent and subsequently subdivided it into individual functional program agents, ranging from 1000 to 13.000 m^{2,} ordered in a Fibonacci sequence. These

Img. 14. Left: ONL[Oosterhuis_ Lénárd] Speed and Friction project, behavioural and physical models.

program agents were subsequently given additional parameters; function type, floorspace area, number of floors, average floor height, additional shape defining parameters and connections to other program elements.

The ability to achieve holistic design quality and exceptionally efficient, yet flexible spatial organization of the project were both high-priority aspects of the masterplan. The project was meant to become a global landmark location, visible from most airplanes approaching to land on the Abu Dhabi international airport. The organisational principles, once again were introduced in the form of “powerlines”, where individual powerlines were treated as agents in the system. This time powerlines had more than an aesthetic role. They served as guiding lines for the building form, but they also defined the internal routing for cars and pedestrians through the designed complex. The powerlines performed as curvilinear repellents for the program agents, resulting in stabilising of their clustering in areas between the powerlines.

The program agents, apart from position and floorspace, were given additional parameters including height, orientation and proportion. In this way, their orientation was always tangential to the powerlines in vicinity and stacking of functions became possible. The initial interaction with agents was mostly performed directly by prof. Oosterhuis; however following detailed interactions were then taken over by other members of the design team, while verbal communication across the design team and frequent team discussions were continuously maintained. While the development progressed, the author kept changing minor aspects in agent’s behaviour, when e.g. behaviours caused local contradictions and positive feedback loops between agents kept pushing the system into a chaotic state.

On the outside, building skin was formed parametrically by project architect ir. Gijs Joosen following a different type of Powelines, in order to achieve its unique articulation embedded in the skin covering the entire building. In result, subtle ripples of the building skin system formed a new landscape emerging from the desert, with most articulate parts covering the hotel and conference centre and voids from which thrives the greenery of newly created oases within the building’s voids.

The programmatic model has remained flexible throughout the process. On several occasions prof. Oosterhuis asked the clients for the feedback on the project and their response resulted in local changes of program distribution, which then resulted in instant adjustment of the building skin and floor slabs, without compromising the local relations between elements. In case of more radical changes the system would become chaotic for a short moment and eventually stabilise itself in a different configuration. This would in some of these cases require slight adjustments in the parametric model, which has a fixed topology, top-down predefined by project architect ir. Gijs Joosen.

c) Evaluation

Despite following similar design methodology, the Speed and Friction project has greatly improved on both complexity and adaptability when compared to the SNPC project. The roof structured has been designed to provide far-going formal flexibility, allowing for radical program reconfigurations. The program cells were designed as autonomous units, where hypothetically reconfigurations underneath the roof structure could have been made after project realisation. The satellite pavilions showed how the approach could be used to create more buildings following the same project logic[1].

[1] Tomasz Jaskiewicz, ‘Behavioral Modeling Applied to the Speed and Friction: Automotive Complex Project’, in *iA#3 - Interactive Architecture Emotive Styling*, ed. Kas Oosterhuis and Xia Xia (Rotterdam: Jap Sam Books, 2010).

6.3. ONL Mnahal Oasis Masterplan project

a) Outset

The Manhal Oasis masterplan has been an endeavour of even larger magnitude than the Automotive Complex. The project was executed in 2006. The project covered a 70 hectares area in the downtown of Abu Dhabi and has been planned to be built up with over 2.000.000 m^2 of mixed function floorspace. The challenge of the project involved the combination of designing of a landmark multifunctional destination for the inhabitants of the entire city with dense packing of functional programme without compromising the appealing aesthetic vision.

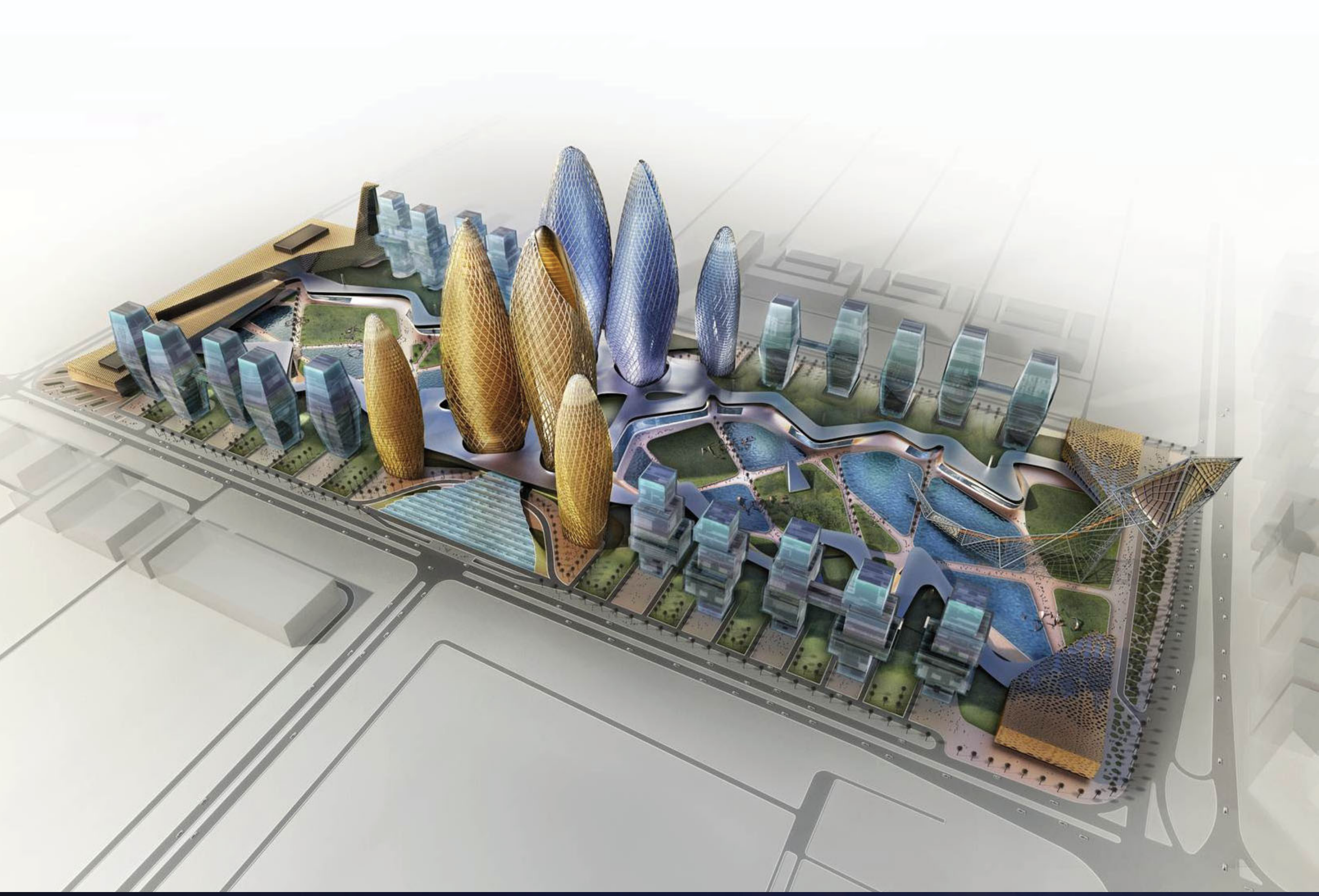
While in the SNCP and Automotive city project, individual program agents were mono-functional spaces, in the Manhal Oasis, due to the scale of the project, only general indications of functions were given in the brief, dividing them into housing, office, commercial, leisure and parking. Thus, the overall program of demands has been defined as a fixed variable and distributed among all defined buildings. The client also supported alterations to the changes in the program, providing that they were supported by feasibility studies.

b) Process

The initiation of the Manhal Oasis project followed a different pattern than the other two projects in this section. The client was a project development firm acting partially on behalf of private site owners. The brief became formulated during a series of meetings between these parties and ONL office partners. Consequently, the masterplan got to be developed inside the network of interactions of the group which formed itself out of clients, developers and architects. Architects took the leading role in masterplan development, but the process involved regular involvement of other parties as well.

The first stages of development of the Manhal Oasis system were similar to the SNCP and Speed and Friction projects. The initially defined functional program, one agent of 2.000.000 m^2 floorspace, became subdivided into generally grouped leading functions (housing, office, commercial, leisure and parking) and subsequently into smaller functional program agents, following the decisions taken by the group. The methodological change implemented in the system was that upon the subdivision, the original agent was not deleted, but changed in type into an “aggregation” agent, while other agents were created with a part-of relation to that aggregation. In this way, the relationship across hierarchies was possible bi-directionally. Depending on specific parameters of agents, the value of the floorspace changed in the “local” program agent would either result in increasing the floorspace of the “aggregate” agent of the functional group and entire project, or the “aggregate” agent floorspace would stay fixed and “local” agents would negotiate among themselves how the locally increased floorspace should be compensated with decrease in other floorspaces.

In the Manhal Oasis project the author took the lead as project coordinator. The multi-agent model remained central in the project development throughout the whole process. During the course of the process the designers, the developers and the clients interacted with the project system, while more subdivisions of agents were made and more relations between agents were established. The project development process became compared by many project participants (including architects, developers and clients) to an intricate board game with multiple players and with complex system of turns. It was a cooperative goal was one uniting factor of creation of a robust, stable state for the entire system with a highly attractive appeal. Throughout the “game” clusters of functional program were formed. These clusters were then grouped again as a new type of agents, each with a distinct name and theme. Some of these clusters were office or housing towers, but most of them were highly



individual buildings, including a souk, two interlocking themed shopping malls, a landmark tower with a viewpoint restaurant, the car park landscaping objects, a museum and a gallery, mosques and last but not least, four large, over 300m high multifunctional landmark towers.

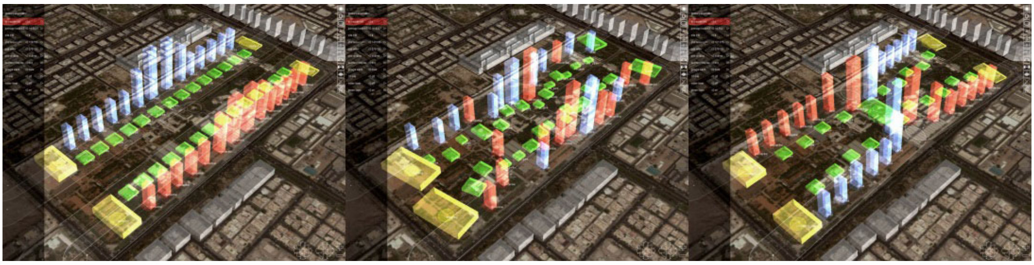


Fig.42. Three configurations of Manhal Oasis functional program assembly during project development

Unlike in the previous projects, there was not one parametric model created following the functional program distribution. A cluster of program agents could be defined by one of the lead designers and then a parametric tower model was assigned to that group, creating a more detailed tower geometry based on the parameters and relations between program agents (Tim McGinley added a number of office and living towers created with his experimental tower generator tool). For several unique buildings custom parametric models were built by ir. Gijs Joosen. Because of such distribution of the parametric models of building forms, reconfigurations of agents remained fully possible at any point in the project development. The final product of the design process was not one plan (although one version was indicated), but a distributed system of building agents interconnected with each other and forming a stable state. Parameters of these agents could be changed by the client within set boundaries at any point in time after project delivery and the system would reconfigure itself and eventually stabilise in a new equilibrium, adjusting itself to changes in the functional program and spatial configuration.

c) Evaluation

The Manhal Oasis project has shown a different approach towards distributed modelling and related interpretation of architectural spaces. Where Speed and Friction was delivering an architectural mega-structure, Manhal Oasis was aimed to develop a diversified downtown area, filled with high quality leisure areas and diverse, recognisable landmark buildings. Therefore spatial organisation has in SNPC and Manhal Oasis, it has been reorganised frequently throughout the entire design process. In SNPC and Manhal Oasis proximity driven self-organisation played an important role in the process. In Manhal Oasis, other types of relations between functional spaces played a more significant role. Even though not all relations were implemented in the virtual design environment, they were maintained by the design team members.

Unlike in SNPC and Speed and Friction, not one but multitude parametric geometries were used in the Manhal Oasis project. Each cluster of spaces would be wrapped by geometrical form of a skyscraper, landscape element or other architectural form. This approach has allowed more intensive spatial reorganisations of the project until the last days of the design work. This included being able to offer the client group a possibility to make alterations themselves at any moment in time, also after delivering the project. In this way delivered virtual system has remained adaptive throughout the entire design process.

To date the project has not been realised and remains on hold. Shall it be revisited further adaptations could be done to adjust it to the continuously changing demands. Equally, the actual realisation process of Manhal Oasis remains embedded in the system logic, where

Img. 15. Left: ONL[Oosterhuis_ Lénárd] Manhal Oasis design.

various phases in which different built spaces are to be materialised are indicated by relations between them. Consequently, further changes to the project could be implemented to the project after its complete initial realisation.

6.4. Discussion

Earlier sections discussed the development of architectural systems as aggregations of dynamic components and inclusion of virtual and actual user agents in projects. This section has demonstrated a parallel methodological trajectory. Instead of starting project development from smallest components, spaces have been used as different and larger scale components that can be subsequently broken down into consequent subspaces and ultimately further defined through components and populated by users after project's realisation. In smaller projects the concept of space could have been omitted, where 1:1 interactions between components and users were enough to design and build rudimentary architectural systems. However, in large and very large projects such as Manhal Oasis design without consideration for architectural spaces was impossible. Conversely, starting from the space, as the container for interactions of people and components has been a logical choice.

The main difference between traditional architectural programming and used method was attribution of agency and flexibility to designed spaces. Where traditionally the functional spatial organisation would be fixed, here the entire network of spaces was left flexible, both in terms of size of space, its relations to other spaces, as well the type of functional programme attached to it. Spaces would consequently have simple behaviours attached to them, allowing them autonomous adaptation depending on changing local conditions. This has provided a structure that building component systems could further populate and which could be filled in by user activities.

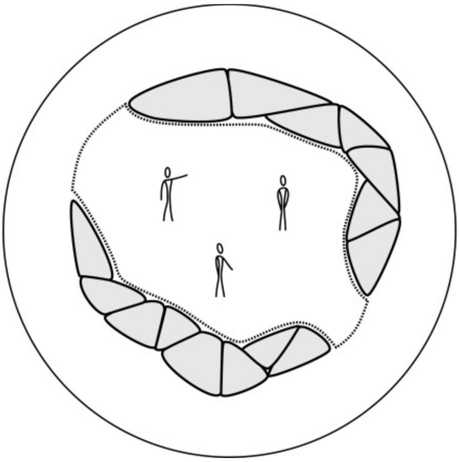


Fig.43. Ideogram: Place as aggregation of components, people and architectural space.

As a result spaces provide the foundation for generation of places; aggregations of objects and people's actions over time with distinct character and qualities. In a place, space would remain the central organisational reference point, providing a location model for local interactions, permitting identification of locality even after complex transformations of the system.

Traced projects have not explicitly focused on maintaining the adaptation of designed systems post the design phase, although such possibilities existed. The problem with continuation of design adaptation in respect to active spaces and beyond the realisation phase is that the building spaces can be easily represented virtually, but don't have an explicit material manifestation in the physical world. After realisation, spaces are defined indirectly by building components and/or activities of people. This means that in order to continue adapting spaces directly, not through aggregations of components or influence on users, the virtual agents need to be maintained in the realisation phase or an alternative method needs to be found for embodying spatial agents.

Conclusions summary:

- Architectural program requires change of focus from permanent function to dynamic activity patterns.
- Agency of spaces can be virtually actualised.
- Places provide a location model for building components and people and permit continuity of systems across stable states.
- Agency of spaces cannot be directly carried on to the physical world.

7. Largely complex formations

Challenge:

- How can complex networks of building components, people and spaces be integrally dealt with?

In projects presented to this point weight was put on agents that are system designers, that are components of material building structures, that represent the functional programme of these buildings or that are inhabitants of created architectural environments. Relations and interactions between the four types of agents were relatively simple, forming sequential chains of dependencies among aggregations of those three types of agents.

The following case study projects are accounts of design processes, where the development of a project occurs simultaneously in all groups of agents. The premise for the following projects was to approach designs as multi-agent systems, where no groups of agents were assumed a priori. Groups were only to be formed as dynamic assemblages of heterogeneous agents during the formation and development of architectural systems.

The most significant of the observed phenomena during projects development were: 1) the formation of the system through local interactions between agents and designers (including clients and developers taking a limited role of designers) 2) the subdivision of agents in the design process 3)the semi-lattice grouping of agents to form aggregate agents of higher order 4) the design activity of stabilising the system at lower entropy levels to counteract situations when system tends to fall into chaotic states 5) the “physical interpretation” of agent aggregates through parametric models and the possible distribution of such process through employment of multiple, small scale parametric systems.

Projects described to this point demonstrate the interplay between building components, architectural spaces and designers during the processes of project systems’ virtual formation. In the projects discussed in the section 1, the system of created spaces and coinciding with them functional programs were relatively simple. There, the spatial organisation was to a high extent already suggested in the project brief. The formation and organisation of building components became thus the primary design activity. The development of building components resulted in thorough refinement of the spatial organisation and spatial qualities, but did not involve changes in its topology.

In projects discussed in section 2, spatial organisation is highly complex. In these projects the topological formation of spatial organisation became the main design challenge. In the design processes topological changes in the networks of functional programme elements were frequent and far-reaching. Proceeding with the formation of building bodies could therefore only advance when stable sets of functional programme formed. Since such stable sets were at times only local, as in the Manhal Oasis project, the assemblies of building components were also only stable locally, allowing large-scale reconfigurations of entire buildings.

In all discussed projects the functional programme and its distribution were designed in detachment from specific activities of inhabitants of the building site. Such simplification is possible when patterns of inhabitant behaviour that lay at the foundation of the project's functional programme are expected to remain unchanged. For example, the Cockpit's function as a car showroom was based on the assumption that the buyers will come to the showroom and generate enough profit for the showroom to be sustained, which proved to be a fair assumption. The internal organisation of the showroom, was based on the company's earlier experience in selling cars which has proven to be successful and there were no reasons to expect any factors that would invalidate its continuation

However, in consideration to the discussion on architectural adaptation presented in chapter II, such assumptions may be risky. In an increasing number of cases, the functional programme evolves and changes based on complex activities and interactions among inhabitants of architecture. In complex systems of functions organised in architectural spaces, such as a fragment of a city, the factors determining the eventual activities that will unfold in these spaces are too complex to be assumed.

7.1. Urban Body - Building Relations

The first experiment organised by the author to evaluate an integrated application of all aforementioned aspects of the multi-agent logic to an architectural design context was executed in the Building Relations design studio co-taught by the author with dr. Nimish Biloria as part of the Urban Body MSc2 course at TU Delft held during the spring of 2006.

In the design studio brief, students were requested to formulate "adaptive designs, to be used not only to create physical (architectural) objects, but also to generate their dynamic interaction. (...) Instead of producing "animated" forms that behave in a linear, thus fully predictable way (...) objects that have a non-linear behaviour (...) [that] adapt and relate themselves to all kinds of complex and unexpected situations."

a) Outset

Eight students participated in the studio and each of them developed an independent project on an individually selected site in Madrid. Before the design studio started, the students had spent a week exploring their design and thoroughly analysing their design sites. The definition of the design brief was part of the design process.

b) Process

The design method was imposed on students. In the initial phase of projects development the work of students involved numerous diagramming exercises of gradually increasing complexity. In these excercises students defined all agents they encountered on their sites, including parts of architectural components, spaces, inhabitants and aggregations of the above, forming "places". Once agents were identified, students proceeded by tracing and mapping relations between these agents, ultimately creating rich networks, subjectively describing the areas of their projects. In the next step, malfunctioning parts of these networks were identified. It was up to individual students to define the problems and opportunities encountered on assigned to them sites, based on the traced by them actor-networks of people, spaces and things (the term actor-network was not explicitly used though). Most identified problems involved the lack of inhabitant activity in public space of the city or activity of only one group of inhabitants repelling other groups. Resulting from that inhabitants of the studied areas did not have ad-hoc possibilities to meet each other, exchange views and ideas, get to know each other better, and engage themselves in shared activities.

The identified problems were observed by students as specific discontinuities in traced by them interactions among inhabitants of Madrid and inhabited by them spaces. The problems and opportunities were then highlighted by students as discontinuities and qualities of these networks, rather than of the static architectural space in itself.

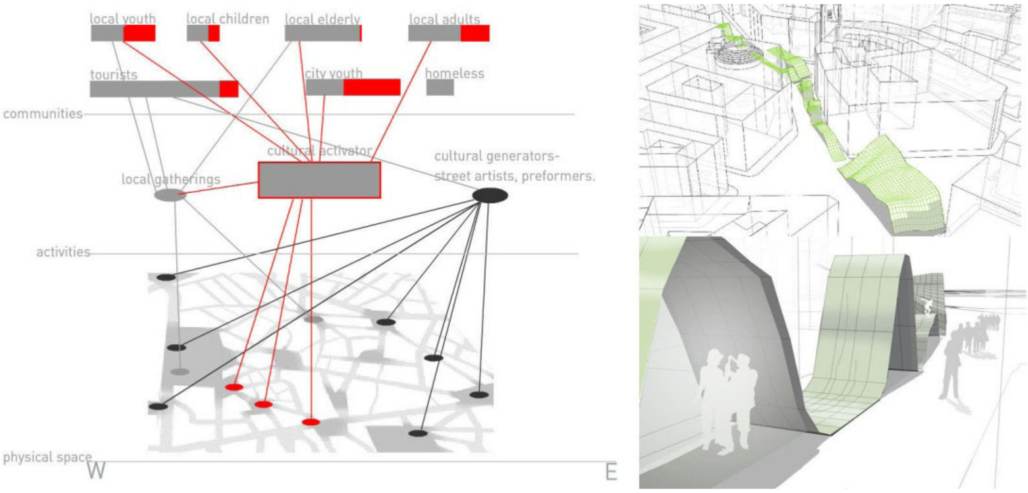


Fig.44. Leisurator, Rannveig Yli

Students progressed diversely from this point on. Rannveig Yli proposed a series of spatial interventions into the public squares of Madrid. The idea for these spatial interventions was to allow combining leisure activities typically performed by area inhabitants that typically don't overlap in space. Rannveig designed a parametric urban structure composed of a long curving strip of varying width could host an outdoor cinema, skate park, and providing various social meeting spots. In this way Rannveig expected to allow people of different social groups to merge and generate a stronger collective cultural identity. Among the projects, Sylvia Rubin proposed a series of differently sized sculpture objects to be placed throughout the city. The size of the objects was parametrically corresponding to the geometry of the surrounding it urban pattern. Lights embedded in objects were in real time communicating the amount of activity of inhabitants around other objects of the same kind. In this way Sylvia expected to stimulate the flow of inhabitants between different city squares and promote exploration of areas of Madrid that currently felt alien to inhabitants of other parts. Marco Boeber redesigned one of the squares into a performance centre, creating a seamless transition between the street space and the interior of the building in order to provide interior space that would feel to inhabitants as a public one.

c) Evaluation

The resulting interventions into the city developed by students were all proposing spatial design solutions that in various ways dealt with the dynamism of urban processes. Some projects dealt with it metaphorically, like in the project by Julia Rubin. Other projects like this of Rannveig Yli, proposed a set of very functional intervention into the city, facilitating activities that were otherwise hindered by the urban environment.

As evaluated by students after the course, the method in which students had to perform opened up a new way of critical thinking about spatial environments. The initial adoption of the method encountered resistance among some students, but eventually all students agreed that the method had greatly increased their ability to improve the city space through design.

Work developed by students showed the unquestionable potential of designing architecture as a rich dynamic system embedded in the urban environment. However, it also became clear that the students were not able to go beyond the conceptual design phase with their skills and design tools they could use. The logics of systems they proposed were conceptualised but not formulated to the level of detail that would have made the inner working of their systems profoundly connected to the complexity of all interactions occurring in rich urban spaces. Students designed features to be inserted into existing urban spaces, rather than thorough reconstitutions of these systems.

7.2. 751 City

The design studio taught in the fall semester of 2006 was from the author's side a continuation of exploration of the design method focused on the shaping of relations throughout heterogeneous systems of architectural agents. The premise of the studio was deliberately more ambitious than in the Urban Body studio, aiming not to augment an existing urban environment but to attempt to recreate the full complexity of a city system using a complex assemblage of interacting agents. The studio was organised as part of the Hyperbody MSc3 (3rd semester in the Master of Science programme) module and was conducted under direct supervision of prof. Kas Oosterhuis[1,2].

a) Outset

Twenty three students participating in the design studio were given a brief to develop a system of an autonomous small-town scale urban environment. However, the given site was not a usual two dimensional plot, but a large, three dimensional urban body. Each student was assigned to one of the 23 interlocking pieces of such 3d puzzle. When put together, all pieces would form a giant sphere of 8.000.000 m3. This spatial boundary has been located in the middle of the 751 factory in Beijing, the eastern part of the 798 art district. The sphere, partly submerged underground, would stand on small feet, a 20.000 m2 base. The rest of the 751 area has been divided among the group of five students from the collaborating South-East University of Nanjing. Their work proceeded in parallel to the progress of Delft students. Their designs, however, have been made from a more conventional, two dimensional starting point.

[1] Tomasz Jaskiewicz and Kas Oosterhuis, eds., *751 - Multiplayer Design Studio* (China Architecture & Building Press, 2008).

[2] Tomasz Jaskiewicz and Kas Oosterhuis, '798 Multiplayer Design Game, A New Tool for Parametric Design', in *Space Time Play: Computer Games, Architecture and Urbanism: The Next Level*, ed. F. von Borries et al., 1st ed. (Birkhäuser Architecture, 2007).

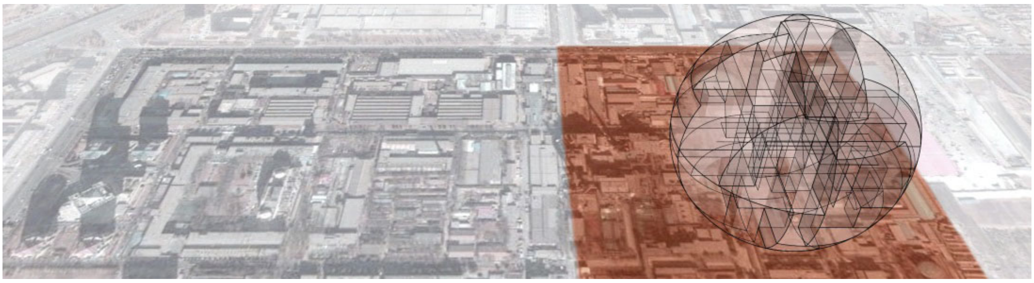


Fig.45. The three dimensional design sectors on the 751 site

Each of the students was encouraged to be creative and innovative; they were allowed to develop any proposal as long as they would obey the rules of the master plan.

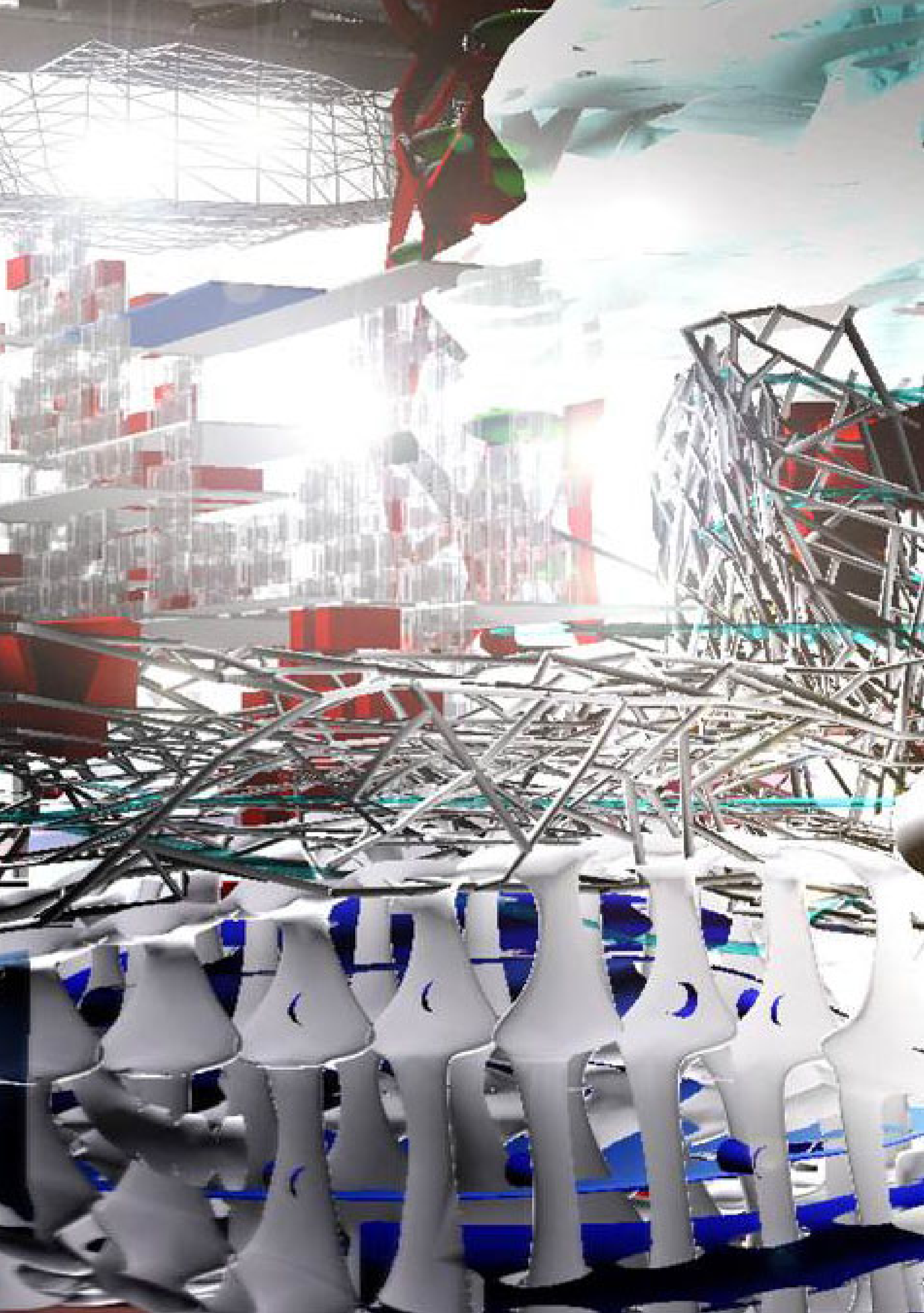
1. Program is mixed use development of 1.000.000m^2 of built-up area.
2. Contained in a virtual sphere of 8.000.000m^3 leaving 25% open space for bringing light into the large urban body.
3. We provide the students with a 3d puzzle of as many interlocking parts as there are students.
4. Each 3d plot communicates and negotiates only with their immediate neighbours.
5. Each piece of the 3d puzzle has a specific program of requirements [housing, offices, commercial, cultural, educational, leisure].
6. Location will be right in the heart of the 751 site.
7. Each plot administrates their data input, data processing and data output and communicates the parameters in a dynamic database with their immediate neighbours.
8. Each plot has to structurally support itself and communicates data of structural loads with their immediate neighbours.
9. The sphere must produce energy as much energy as it consumes.

The rules were formulated based on the principles of swarm behaviour. In a swarm each member of the swarm exclusively communicates with their immediate neighbours. The shape of the swarm is not imposed by any of these swarming members, there is no leader nor do they have an awareness of the whole. The shape of the swarm is the balanced result of the bi-directional interactions between the acting members and of exterior climatic conditions which impose constraints to the size and direction of the swarm. In the 751 master plan the students are the bottom-up communicating swarm members and the tutors represent the top-down control.

The students from the South-East University working on the 751 area around the sphere were asked to approach their designs more conventionally. Their task was to react to the developments within the sphere and to provide facilities for that development while accomplishing their particular design goals and acting as an intermediate zone between the sphere and the rest of the 798 / 751 areas and Beijing.

Three-dimensional plot distribution according to the master planning scheme meant that some designs were located above the others. This implies that some projects had to be structurally supported by the other ones. Also access to plots which were not directly connected to the ground level along the boundaries of the sphere had to happen through other plots.

In this way students could not design without respecting what their neighbours were doing. Design decisions taken by each one of them were always leading to all the other projects requiring adjustments. In their work process students have realised themselves that under



such conditions designing in a pre-defined, fixed way becomes extremely inefficient. The circumstances have forced them to think and work flexibly, so that projects could be changed instantly while their surroundings were evolving.

b) Process

The development of the 751 system began with the pre-arranged subdivision of its spatial boundaries of sectors. The role for these boundaries was to provide a framework for the system to unfold in. Although three dimensional, the boundaries were clearly analogous to real-world boundaries of land ownership, or legal land-use distribution. The students started with an initial indication of the functional program, but they were allowed to trade that program among each other. They were also free to interpret this program flexibly.

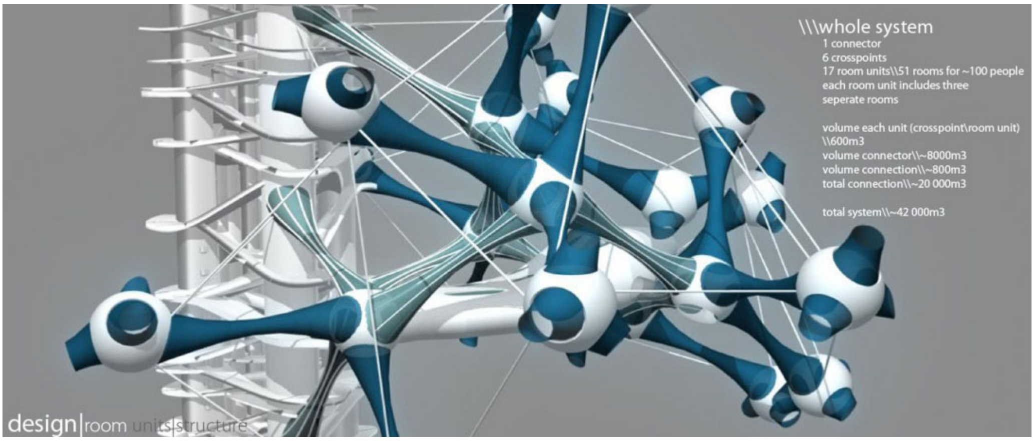

\\\whole system
1 connector
6 crosspoints
17 room units\\51 rooms for ~100 people
each room unit includes three
seperate rooms
volume each unit (crosspoint\room unit)
\\600m3
volume connector\\~8000m3
volume connection\\~800m3
total connection\\~20 000m3
total system\\~42 000m3
design|room units|structure

Fig.46. Module by Roman Krajger attached to module designed by Mirka Tumova

Within the first weeks of project development students came up with first ideas for the spatial organisation of their sectors, dictated not only by their own individual creativity of students, but mostly by the location of the sector in relation to other sectors and by sector's shape. For example Miroslava Tumova, responsible for a vertical sector in the middle of the sphere proposed a large structure that could structurally support other projects and save for vertical communication in the sphere. Wang Lin, whose sector was located at the very bottom of the sphere proposed a waste processing plant with related to it educational facilities. Roman Kraiger's sector located on top of the sphere became a combined hotel and residential zone sue to its breath-taking vistas. These ideas were only allowed to be considered once the students found others to exchange their originally assigned program of demands with.

This was only the initiation of the design process, because following the initial ideas, students had to make sure that the 751 sphere would work as one, complex, urban ecosystem. For this, in subgroups they devised methods for rapid exchange of information about exchanged structural loads, energy and human flow, water, sewage and waste throughout the system. On one hand students had to develop the designs for their sectors in detail. On the other hand they had to continuously adapt their projects, as change in one sector, would result in requirements for changes in other sectors.

Many sectors were developed as multi-component systems to permit such flexible adaptations. Some projects, such as Dada Wang's water-tower, a joined initiative of Jenna Fitzel, Felix Wurs, Isil Sencar and Eva Kiesel called "Hive Mind", and the project of Lucas Mahlknecht involved different sorts of mobile pods that could travers parts of the sphere ecosystem. Although the flow of users through the entire sphere was not simulated, user behaviour and following it

Img. 16. Left: 751 project, view inside the collaborative model.

uncertainties was taken into account. Many students used personas and imaginary tracings of such personas interactions throughout the sphere in order to justify many of their design decisions.

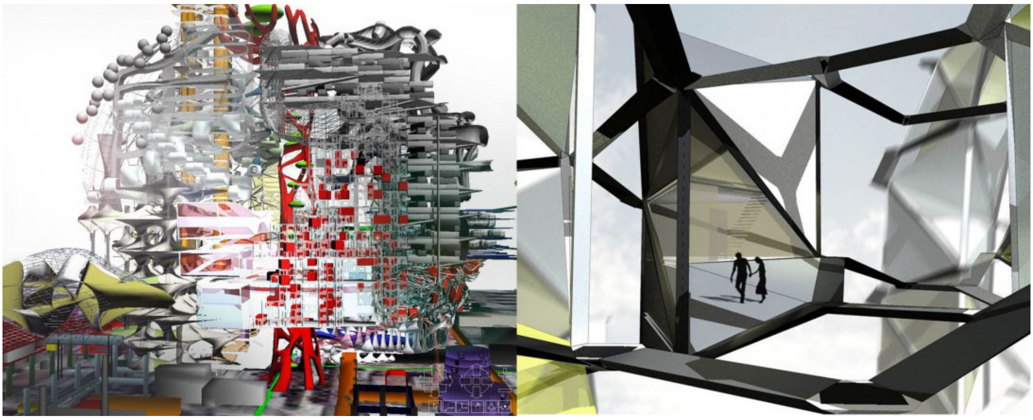


Fig.47. The complete assembly and zoomed in fragment (sector of Alexander Baum)

The complete 751 project emerged as a system of systems, in which individual sectors were heterogeneous multi-agent systems with rigidly defined sector boundaries. The sector systems were continuously exchanging forces, energy, material and people between each other. This led to their continuous transformation. The virtually constructed 751 project never stabilised. It continuously developed and transformed until the course came to an end. In this it greatly resembled actual city systems, which develop continuously. However, in the 751 project the development was much faster than in actual cities.

c) Evaluation

The final result of the design studio has been a highly complex system. All designs assembled together formed a giant structure. If built, this structure would have all the qualities of a small city, but instead of being spread over a two dimensional ground surface it would function in three dimensions on all detail levels.

Even though they were forming one large entity, each of the sector designs has maintained a fully unique character and organisation. Some were embodying well defined, fixed architectural spaces. Other ones were flexibly responding to altering demands of their users. Many of them consisted of a high number of cellular elements, while others were just singular bodies embracing all inner spaces under one skin.

Although the data exchange between designers has not been real-time and often was achieved with primitive technical means, the outcome of the studio showed great potential for a true multidimensional interactive urban design. It has proved that designing with instant communication between members of the design team provides far more robust results than doing the same work in sequential steps. In this particular case each of the 23 designs became one member of a swarm forming a whole, which is much greater than just the mere sum of its parts.

7.3. Distributed faculty

The 751 project was developed in a methodological framework that proved generally successful. However, the inclusion of the rigid subdivision into sectors became a substantial constraint in system development. The project sectors were assumed arbitrarily upon the start of the project, clearly imposing a constraint that might have ruled out a broad range of project variants. The other constraint assumed from the start was the global functional program constraint. In the development of any architectural space, the functions can change based on the actions of inhabitants and hence the balance of functional square meters is bound to be altered. The brief for the Distributed Faculty Design studio, held in Hyperbody MSc1 semester in 2008 was devised to develop architectural systems analogically to 751, but with less constraints.

a) Outset

The challenge for the combined MSc1 and MSc3 design studio was to design an architectural system that would replace the faculty of architecture building which burned down in the spring of 2008. Each of the students was obliged to design a different component of the faculty in a flexible, dynamic and parametric manner and as a multi-component assembly.

MSc3 students were asked to design a variety of systems forming the "bones" and "skin" of the new faculty, while MSc1 students had the assignment to design "organs" of a distributed faculty of architecture. Each "organ" was to be an autonomous spatial installation, allowing accommodation of a specific range of activities. Each of the MSc1 students had to design two building "organs". For the first one, students received a selection of activities to design for. For the second one, they were free to choose a purpose by themselves.

The functional brief was left open to the students. Instead of designing for specific functions, they were requested to identify activities normally taking place in a faculty of architecture and design architectural spaces that would best accommodate these activities, consequently assembling and moulding these individually created modules into larger assemblages, forming eventually the entire faculty.

b) Process

The project has started by each of the 18 students interpreting two specific activities taking place at the faculty of architecture. Activities included belonged to one of the six categories: learning, designing, socialising, researching, supplying and organising. Those activities were taken as points of departure for defining spatial affordances and qualities for the initially individually developed modules of the complex faculty building. Students were free to decide upon the scale of their individual modules and could opt for a module to become spatially distributed throughout the faculty.

Initially students progressed individually. Modules were developed as standalone systems. All considerations for the exchange of people, forces or energy through module boundaries were assumed generally. In the process of developing their modules students employed different strategies. Most of them started by a thorough persona-based analysis of activities occurring in selected categories. Such hypothetical personas were then employed conceptually as virtual agents triggering or requiring specific affordances of space occurring in a number of interwoven scenarios. The formation of virtual component assemblies followed. It was performed by devising parametric systems, which would govern the process of populating the module space with unique material components and which would provide affordances required by identified activities and related small-scale spatial organisations. Additional activity types were introduced by some of the students. In parallel to that, master 3 students designed four variants of building component based envelope systems for the faculty – its "skin and bones".

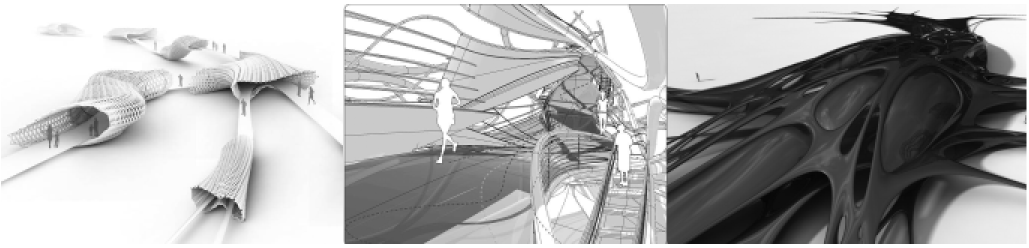


Fig.48. parametric components for the faculty of architecture, authors: Agata Kycia, Jonas Sin Po Sing, Bao Ang Nguyen Phouc

The modules were constructed flexibly, in order to allow further assembling of modules into larger-scale systems. In due course students were asked to combine individual modules in various ways into larger systems and chose best of such combinations to ultimately serve as official entries for the international design ideas competition for the new faculty of architecture building.

In the process of combining the individual modules of the students, students formed four smaller groups, each including several members from both MSc1 and MSc3 studios. The goal was to combine the different "organs" with the "skin and bones" of the building as well as with the dynamic activities of real and virtual users of the faculty. Additional task for the "skin and bones" was to serve as an interface to the outside world, allowing individual projects to relate to the environmental conditions, regulating access to the building, providing structural support along with many other factors and features.

During that stage of the development process, the 5 groups of students created 5 proposals of assembling these components into working wholes. Resulting designs were submitted to the official ideas competition for the new faculty building. This was a side track of the design studio, but considered a good exercise to deal with deadlines and actual requirements set in a design competition. Most of submitted ideas envisioned the future faculty as a dynamic system, which can in multiple ways evolve and change over time, while adapting to changing demands and external constraints. Parametric way of thinking and designing, present in the studio form the very beginning facilitated this kind of approach.

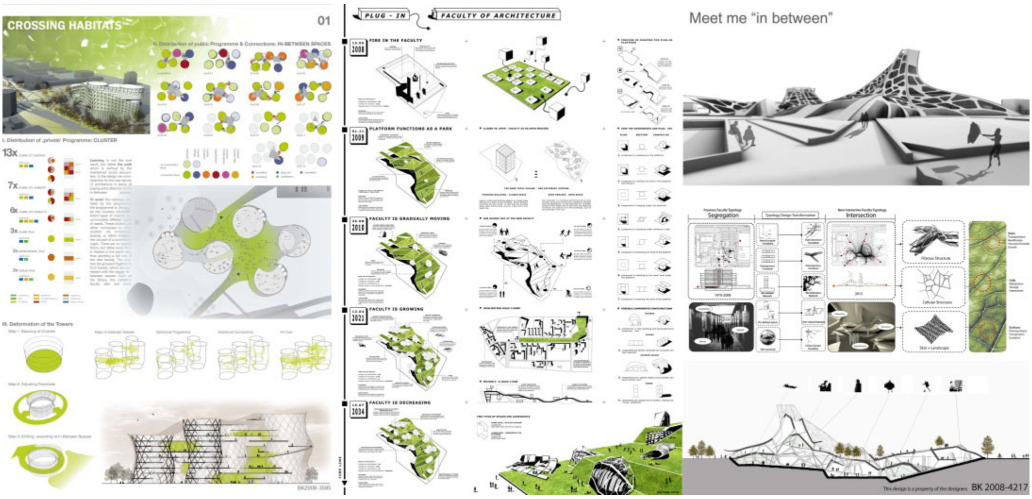


Fig.49. selected panels submitted to the building for bouwkunde competition

The ultimate challenge for the students was not only to develop a valid design, but also to make it successfully perform in the complex and dynamic project environment. For this, each of the students had to closely collaborate with other students, making sure that their projects work together as robust ecosystems.

In the last phase of the studio, student projects were developed further to a highly detailed level, proving that the most unusual of proposed design ideas would be feasible to be realized in practice. The challenge was to create technically sound details for projects that in many cases were not static. Several students chose to solve this problem in a systemic manner, coming out with algorithmic solutions that could be applied to varying conditions and produce changeable technical solutions.

c) Evaluation

Designs created by students came together in multiple configurations to form truly complex spatial systems. Faculty components designed by the students were created flexibly, in order to be applicable to diverse and often also dynamically changing contexts. Seen as a whole, this collection of designs has a potential to be combined together in endless variety of configurations and scenarios, to form rich and robust innovative premises for the future faculty of architecture. Some of these configurations have been shown in the ideas completion entries and published in the "building for bouwkunde – open to ideas" book along with the exhibition at the Dutch Architecture Institute (NAi) in Rotterdam.

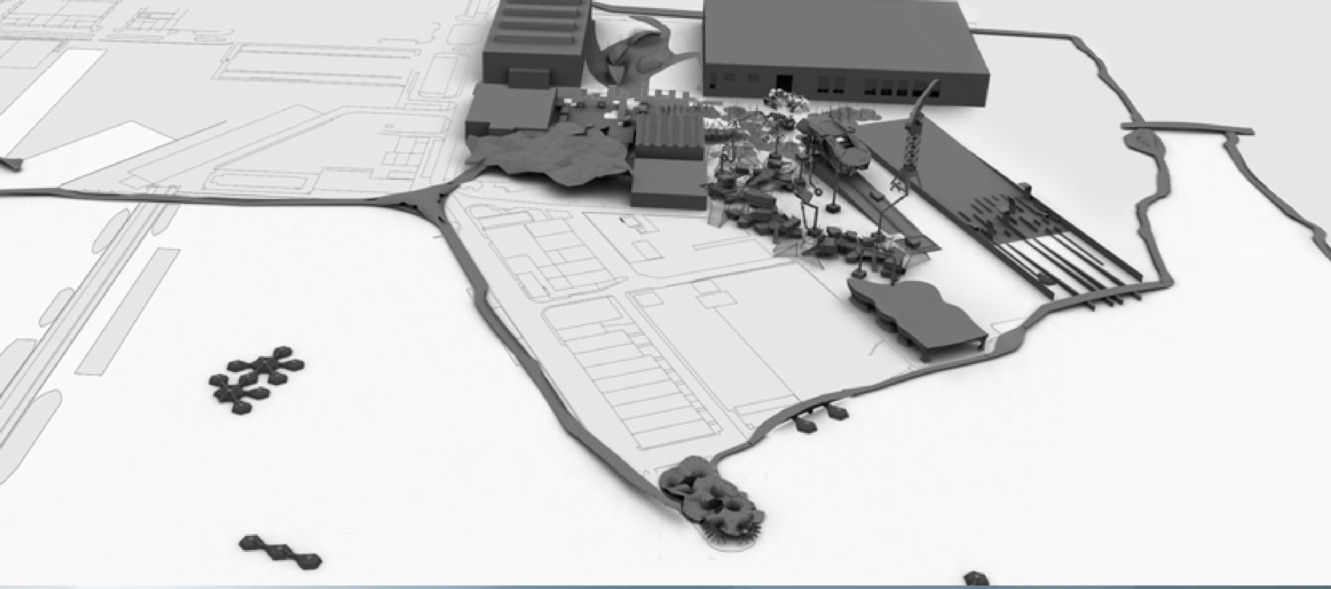
The formation of systems did not follow any predefined topology or linear progression between different types of agents. Individual modules were from the start developed as heterogeneous systems of building components, spaces and virtual inhabitants. The assembly of modules into large scale systems formed new spaces, increased the complexity of relations between module building components and created new insights into activity patterns of inhabitants to unfold in developed spaces. At the same time the loosely defined modules became stabilised by the constraints coming from the entire assembly.

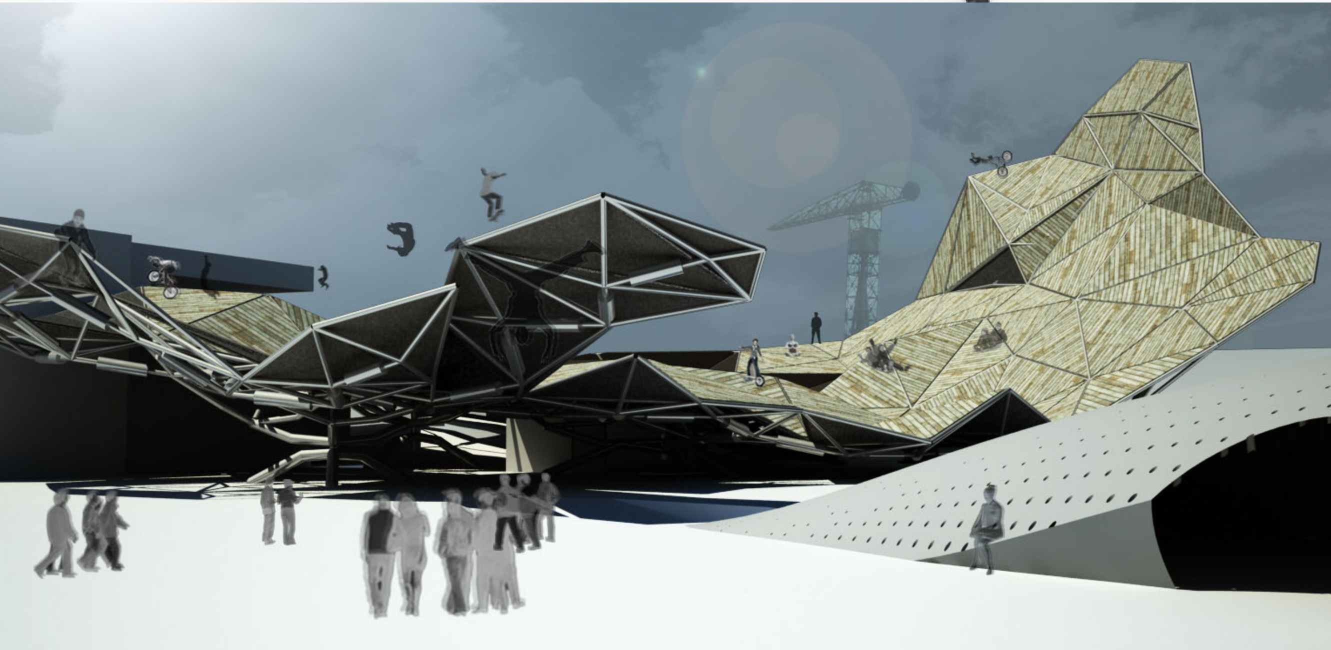
The design studio has demonstrated that a distributed approach to design can lead to radically richer and more complex design results in a much shorter time than a typical, linear design methodology. Developed projects show unprecedented qualities and create often surprising, yet highly functional spaces. Most importantly, however, the distributed approach to design allows for extensive reconfigurations and adaptation of projects on local and global scales throughout the entire project development process. The distributed faculty project has shown how quickly and efficiently radically different and complex projects can be assembled out of autonomous, adaptive modules.

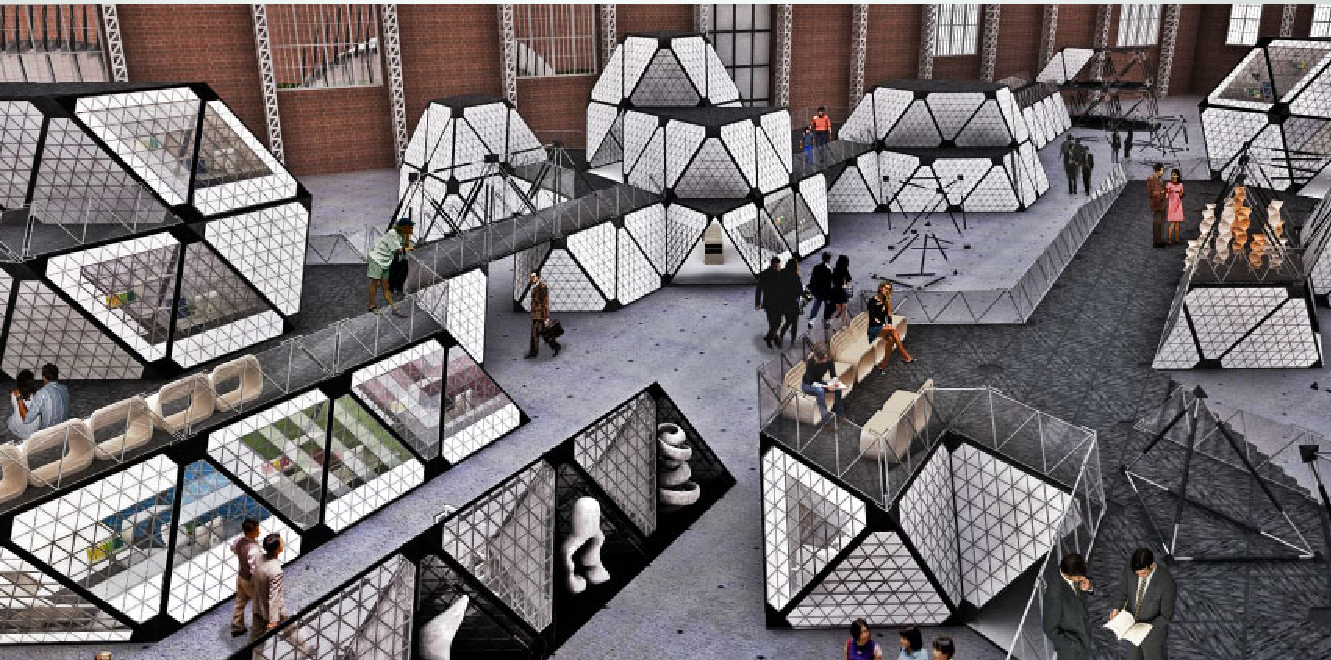
7.4. reNDSM

a) Outset

The reNDSM project has been organised in order to combine the general design approach initiated in the 751 city project with more specific design methods developed in later projects in conjunction with direct connection of the project to the complex real-world setting, where also direct involvement of validators could be included. The project has been a Hyperbody MSc1 assignment given to a group of 18 students in the spring of 2012. Project's location has been the NDSM area in Amsterdam. The site has been selected due to its unique character and multi-layered social and spatial complexity. Formerly a ship-building area, since 1990s NDSM has been growing as the creative hub of Amsterdam. Abandoned shipyard warehouses were initially squatted by artists and eventually with municipal support became organised into a creative colony, with tens of art, design and fashion offices occupying the NDSM loods. These







activities in conjunction with good connectedness to the city centre have not only started attracting organisation of many mass entertainment and cultural events to the area, but have also led to many companies, such as MTV networks Europe, Red Bull or Hema to open their offices within NDSM perimeters. Despite the recent commercialisation of NDSM, the bottom up, do-it-yourself spirit prevails among NDSM users.

The reNDSM project has been initiated by the author in conjunction with prof. Kas Oosterhuis' guidance and co tutoring of Chris Kievid. The idea of NDSM as project location emerged out of earlier collaboration with, among others, Lilet Breddels, director of Archis foundation, Vincent Schipper and Alexander Zeh, and was given support by NDSM foundation's director Frank Aalsema. This collaboration provided the context for the assignment which stemmed from an actual need to provide a counter-proposal for a traditionally planned, top-down masterplan for the NDSM area. The ambition has been to show a strategy for sustainable growth of NDSM, building up on its creative potential and current vibrant and diverse user community.

b) Process

Students started their designs by working individually. In the initial phase the main goal given to them was to identify the key actors in their area of choice, either being people, buildings or things, and through interactions with these actors uncover the way in which NDSM operates, building up a network of interconnections and dependencies between discovered actors. For this, each identified actor was labelled with a unique "protoTAG" (see IV.3.8) and through online interface information about this actor and relations to other actors were added by the designers or the actors themselves. Yet, despite more success than in the VHpark project, feedback from site inhabitants was not gathered with high success rate and would not persist throughout the entire semester.

Individual interventions were proposed by students in parallel to each other. That meant that instead of proposing alternatives to future development of the area, all projects could potentially be realised next to one another. The overarching ambition of the assignment has hence become to deliver a proposal for a future NDSM area's growth as an ecosystem of people and architectural spaces. Instead of top-down planning, the focus has been put on engagement of site's current users and attraction of new users. The projects consequently aimed at providing a variety of architectural means to provide spatial affordances for resulting intensification of use, as well as creating pro-actively spatial incentives attracting new users to the area and positively influencing current users to contribute to site's growth and its flourishing.

Initial projects included a broad range of interventions. Numerous projects proposed various systems for "self-building" of commercial and office spaces using various technological approaches. In that project varied in respect to control given to the users. "Modular customization" provided a kit of parts that would allow users to build any kind of structure, only providing feedback and reactions on structural stability of the system. On the other hand "Interactive garden" project proposed fully autonomous outdoor elements that on their own would reconfigure responding to sensed usage and climate parameters, giving little direct control to the users. Other projects such as the "Quake/X" skatepark, "de Markt" retail area or "stigmergicscape" multifunctional zone worked with a mix of user control, top-down design and bottom-up emergent behaviour of the system itself.

The second phase has been focused on prototyping. Students, either individually or in small groups, explored techniques needed to realise their projects. It has been a reciprocal process where prototype findings would feed back into designs, while designs were guiding the prototypes. Prototypes included physical tests of mechanisms and material properties, computational algorithm for form finding and optimization, project interfaces, and others.

Img. 17. Left, from top: reNDSM collaborative model, Quake/X project,
Modular Customisation project.

In the following last design phase three pairs of students have decided to join their individual efforts and merge their projects. Other designers chose to continue working individually. In respect to this, tutors' guidance enforced connectivity between projects. This has initially met with resistance from students, seen as conflicting with their individual project ambitions. Ultimately all students have recognised the relevance and opportunities of connections between their projects and attempted to take advantage of other designer's projects developed for surrounding areas. As a result a number of studio debates focused on the global project organisation. Students realised that they in fact need to agree on a political system required to be put in place, where some were in favour of top-down dictatorship of operation rules, while others postulated highly liberal or even entirely anarchistic solutions. Ultimately a distributed system prevailed, where each project's designer was top-down responsible for setting rules on her own project, while connection zones were governed by consensus. Clearly, this system would have not worked in realised projects, where actual users would claim their rights and ownership of designed projects and municipal and national regulations would be enforced.

The final presentation of student projects has been attended by a number of NDSM users and professionals with interest in the area. The reception of projects has been highly positives, with few critical comments. Despite the highly non-standard nature of projects, they have been considered as feasible interventions to the area, including NDSM foundation's director plea to the students to continue developing the designs and attempt to develop some of them through bottom-up fundraisers.

c) Evaluation

The project has revealed the potential of distributed multi-player and multi-stakeholder design. Projects have clearly proposed a valid and attractive alternative to top-down urban and architectural planning. However, as students have indicated themselves, the involvement of actual site inhabitants has not been satisfactory. The protoTAG system and iterative online publishing has not provided enough incentive for users to actively give feedback to projects. Due to strong project moderation by tutors, projects have developed strong connections and interactions to each other. Similar connections should have been developed to existing buildings, people and artefacts.

Most students have succeeded in proposing economical models for projects. Fewer have shown project's capability to evolve and adapt over time beyond designed constraints. However, when re NDSM is considered as a whole it demonstrates an unprecedented degree of complex adaptivity. Individual projects have functional overlaps. Therefore, if realised, they would compete with each other, ultimately leading to further specialisation, improvement, relocation or destruction of individual projects, but in all cases advancement improvement of reNDSM seen as a whole.

7.5. Discussion

The 751city and reNDSM show similar, highly complex design processes, yet approached from different angles. 751 project has been designed using a set of clear, top-down defined rules. The three-dimensional site sector boundaries were fixed. Projects would communicate with each other only through the connecting surface. There was little connection to site context and its inhabitants, despite the SEU student projects potentially serving as an interface between the sphere and external city.

The reNDSM project introduced soft project boundaries and project locations were flexible. As a result some projects changed location several times as they were designed. This has provided for more adaptation possibilities. It can be envisioned that after realisation, some of the projects could remain mobile and gradually colonise or abandon different areas of the site, based on local conditions and interactions.

The other advantage of reNDSM was the involvement of local "validators". The critique of the 751 project has been its disconnection from the locality of Beijing's creative 798 district. In reNDSM projects directly evolved from similar locality of the NDSM area.

Yet, from the methodological standpoint reNDSM is a direct successor of the 751 city. The main advancement is the shift from a static to dynamic network of projects and their sub-components. The network, structured "horizontally", that is without any imposed hierarchy proved to be an accurate depiction of the complex reNDSM ecosystem. However, such network has also proved to be difficult for designers to work with. Very quickly the number of individual components and connections between them became too high to deal with. Provided design instruments such as protoTAG and its interfaces proved useful, but not enough to solve the problem of comprehensively managing the projects' complexity.

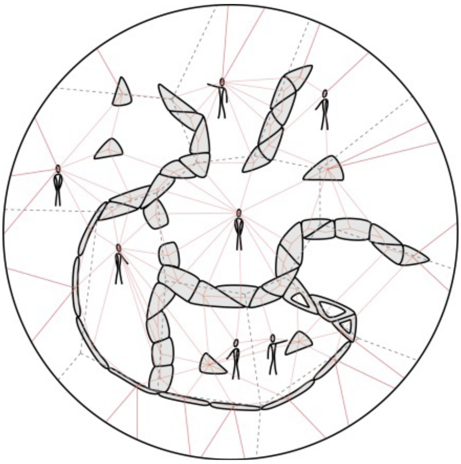


Fig.50. Ideogram: Integration of multiple components, people and spaces through a network of relations.

Conclusions summary:

- Highly distributed design offers a highly viable alternative to top-down design and planning of complex projects.
- Networks of relations provide means to integrate architectural system components.
- New instruments are needed to deal with highly complex networks.

8. Towards in-system design strategies

All projects whose development was traced in section 1 were dealt with architecture as a complex adaptive system. In each of the described projects all project components and participants were treated as autonomous agents. The projects provided insights into complex interaction patterns, among their virtual and actual material components, architects, clients, other experts as well as inhabitants. The development of projects involved gradual enhancement of complexity of developed systems while maintaining their stability and robustness. Each project has been developed not as a model of to-become reality, but as an actual architectural system, being formed and developed through interactions and eventually going through a bifurcation from virtual into a physical architectural space.

In the first group of discussed reference projects, the focus was given to assemblages of material building components. Subsequently agents of functional architectural "spaces" were introduced and the interplay between matter and space and resulting from such interplay affordances were explored. Eventually, the role of human agents was brought into the focal point of case study experiments. Those human agents were either designers or inhabitants but the distinction between these two types eventually dissolved. Once all project agents were discussed separately, more complex patterns of architectural system formations were traced. Consequently, paths for transforming architectural system development from purely virtual to physical, actual operation were presented and discussed.

Throughout all design research case projects, it became apparent that the greatest part of design challenges was to stabilise the systems and prevention from their bifurcating into chaotic states as the systems grew in complexity. Positive feedback loops were avoided. The general goal of involved designers was to maintain a state of low entropy among constructed by them system agents and their assemblages and devise ways for that entropy to be further decreased. Harmoniously distributed low entropy in systems went in pair with robustness of these systems.

Heterogeneity of agents has been recognised as a quality in developed systems. Starting from uniqueness of all agents treated as belonging to the same kind, to understanding that systems require interactions of agents of different kinds in order to maintain their robustness. Among interactions of agents of different kinds, interplays between spaces and material components became critical. It was shown that adaptation cannot be achieved in a linear chain of development from human decisions, through formation of spatial functional programme, to component distribution. Horizontal interactions between all these kinds of agents were considered essential for global and local adaptations to occur in ways that maintain project's robustness.

Consequently, the separation between designers and inhabitants dissolved. Designers traditionally act top-down on architectural systems while inhabitants use these systems and interact with them in a bottom-up manner. In evaluated processes, the distinction of such bottom-up and top-down interactions lost its relevance.

In some of the projects designers-inhabitants were treated equally, they all assumed same roles. In other projects they were given different roles based on their. When designers-inhabitants had to deal with simple problems, equal distribution of tasks was most effective. When more complex problems were at play, only expertise-based division proved efficient.

The distinction between virtual and actual agents remained very strong. The word "model" has not been used above descriptions, however in many cases virtual agent systems were considered by designers as representations of the actual reality, rather than its extension. A provisional conclusion can be made that this results from the earlier education and design tradition. The distinction disappeared with the moment that designers became involved in material prototyping. From the moment that a virtual agent became connected to the physical one, it was commonly considered as part of the physical installation, not its representation.

Technological aspects were deliberately removed from all descriptions. However, technology became the bottleneck in many of the projects. Many features of the virtual agents were dictated by the technical limitations of the employed software. The performance of material agents was limited to sensors, actuators and microcontrollers available at hand. However, most significantly the lack of easy convertibility between physical and virtual agents became the main bottleneck in integrating the two worlds. Following sections will describe in detail the technical developments in providing and improving a virtual environment for construction of discussed systems and technical developments that ultimately allowed physical construction of multi-agent interactive objects. These investigations will lead into prospective convergence of both kinds of systems.

V. Tracing evolution of design instruments

Summary

Chapter V investigates design case study research projects from the perspective of employed design instruments and their role in the design development process. The chapter shows how design instrument prototypes have been evolving alongside the developments in design methods and how reciprocally those methods have been affected by availability or lacks in the instruments.

Various features of instruments are discussed and analysed. Initial set of function-specific instruments is consequently replaced by an ecosystem of instruments which hosts the virtual development of the designed system.

The chapter starts with a discussion about the role of an instrument in the architectural design process and the agency of instruments (section 1.). Following that discussion development of instruments adhering to earlier presented case studies is presented chronologically. First a number of initially dispersed endeavours are discussed (section 2.). Secondly an integrated ecosystem of protoKIT instruments is discussed (section 3.). The discussion points out the shift in approach from building multiple instruments to work on one virtual reality environment towards creating an ecosystem of non-hierarchically organised instruments. Eventually it also raises the question of the transition between virtual design and project realisation, hence providing an entry point to chapter VI.

1. iA design instruments

Continuous exploration of new methods, tools and techniques, experimentation with creation of prototypes of embedded agents, their behaviours and affordances, all leads to increase in scale and complexity of created interactive spatial environments and provides foundation for development of complete complex interactive buildings. Developing interactive designs and installations in an iterative adaptive manner has empirically proven attainable.

As explored in design research case studies, behaviour of individual agents can be described in words, illustrated with flowcharts and even acted out by designers themselves as in the “wizard of Oz” technique[1]. Subsequent creation of working prototypes of such embedded agents can be done with relative ease and will be further discussed in more detail in the next chapter. However, true complexity and resulting complex adaptation in architectural systems means development of systems constituting of a high number of agents, where also some of the agents can be specific to the design instruments (e.g. as “intelligent design objects” defined by Bittermann[2]). Behaviour of such system as a whole cannot be easily extrapolated from behaviours of its individual components. Therefore, as the number of considered agents

[1] Nigel Cross, *The Automated Architect* (Methuen, 1980).

[2] Michael S. Bittermann, I. Sevil Sariyildiz, and Özer Ciftcioglu, ‘A Computational Intelligence Approach to Alleviate Complexity Issues in Design’, in *Complexity Theories of Cities Have Come of Age*, ed. Juval Portugali et al. (Berlin, Heidelberg: Springer Berlin Heidelberg, 2012), 347–368.

increases, it becomes essential to work in an environment that allows rapid development and deployment of diverse kinds of agents, their behaviour and performance in a large system before the physical prototyping can begin and with numbers and kinds of agents that cannot be physically prototyped.

A virtual model of an interactive architectural system, despite consisting hundreds, thousands or more agents can operate on a single personal computer. Various programming frameworks (such as Java-based JACK, JADE or JASON -interpreter of the agent-oriented programming language AgentSpeak or easy to use NetLogo) facilitate creation of software agents and virtual deployment of multi-agent systems. From the architectural design perspective such frameworks are not sufficient. This is due to several encountered problems, such as:

- in-depth programming skills are required, which architects commonly lack,
- traditional computer programming typically require clear a clear a priori definition of system's boundary and requirements
- user engagement in system development is difficult
- development of system through gradual addition and specification of previously unknown parameters is not possible
- systems are limited to computational agents, other agents such as humans or objects without embedded computing devices are considered outside of system's boundary

On the other hand, as discussed in chapter III, section 3.3, design tools contemporarily employed in architectural design cannot fulfil this task either. Clearly, new instruments are needed to facilitate creation and further experimentation with Faced problems.

In his manifesto to designers Bruce Mau has written: "Make your own tools. Hybridize your tools in order to build unique things. Even simple tools that are your own can yield entirely new avenues of exploration. Remember, tools amplify our capacities, so even a small tool can make a big difference"[1] This thought of Mau has become the inspiration in the iterative process in which throughout years of research the author has investigated paths of creation of virtual environments in which architecture could be developed as a complex system of adaptive, interacting with each other agents. Yet in respect to architecture, and in the context of this research the word "tool" should be replaced by the notion of an "instrument". "Tools" are means used to facilitate a repetitive and predictable activity and are meant to be used in one, specific way only. "Instruments" on the other hand are meant to be engaged in an interaction with the artist playing them, where the final creation is produced out of that reciprocal process. Similarly, instruments developed in the presented research have had a strong influence on the projects and designers engaged in the process and were in turn influenced by what was created with their help.

This development has been parallel to earlier traced design case study research and various prototypes of said instruments have been applied in earlier described projects. The instruments have been evolving across those projects. Consequently a separate tracing can be made focusing on the development of instruments rather than specific designs.

[1] Bruce Mau, 'An Incomplete Manifesto for Growth', 1998, http://www.brucemaudesign.com/manifesto.html.

2. Constructing a collaborative virtual design environment

Challenges:

- Commonly used design instruments don’t support design of dynamic buildings.
- Commonly used design instruments don’t support design of buildings as multi-agent systems.

The starting points for the development of the new instruments for architecture have been summarised in chapter III, 3.3-3.5. Clearly, new design methods require new instruments. The critical problems are the enabling of virtual formation of multi-agent architectural systems and participation of multiple designers, experts and users in the process of development of these systems. The Protospace laboratory has been initiated by prof. Kas Oosterhuis at the Hyperbody group with those two purposes in mind and the author has been involved in its development since its early days.

2.1. Protospace 1.0

a) Outset

The ideas of creating a design environment for multi-stakeholder participation can be traced back in the work of prof. Kas Oosterhuis and ONL to the 1990s. Projects such as Poly-nuclear landscape, Parkcity Reitdiep or Variomatic are among more examples where designs took place in virtual environments and multiple experts, stakeholders and inhabitants could participate in the process of formation of the design. Protospace has emerged out of these ideas as a professional research laboratory aimed to develop such instruments further and enhance their capabilities[1].

b) Process

In 2001 professor Kas Oosterhuis has proposed a concept of what was originally called the A-cave. It was in initially conceptualised as immersive virtual reality space surrounding people that enter it with rich experience of a virtual world provided by all-round displays located on the space’s walls, floor and ceiling. In this space, university students and designers were expected to experience virtually formed architecture as if they were inside designed by them real buildings.

This concept continued to evolve further, stimulated by the new opportunity which came up for the university of Delft. The agreement was made to buy (for a symbolic price of one euro) the state-of-the-art example of non-standard architecture, the Web of North Holland pavilion structure. This architectural object was designed by ONL [Oosterhuis_Lénárd] and built for the Floriade exhibition in 2002. As the exhibition was over, it was decided to rebuild the pavilion in front of the faculty of Architecture in Delft. Marking its new identity at TU Delft campus, the pavilion has been renamed to the “iWeb”.

The intention for the function of the iWeb was to adapt it for permanent housing of a collaborative design and research system. This was to be achieved by means of embedding an immersive virtual setup inside of it. The aim of this space has been to allow the development of prototypes of spaces, hence the name of its function which came forward quite naturally:

[1] Kas Oosterhuis, ‘Protospace 1.0’, in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012), 516–532.

Protospace. The design activity in Protospace has been originally described by professor Oosterhuis in the following words: “While designing in real time, the players will open up their sources of knowledge to the others. Protospace is an open source design studio for Rapid Protospacing. Rapid Protospacing means the making of fast and informative prototypes for organizational diagrams, spatial designs, planning schemes and project development concepts. Working in Protospace means augmented insight within a condensed timeframe, and hence more value for money, especially in the conceptual stages of the design.”

The project plan was further developed by dr. ir. Hans Hubers in cooperation with ONL, ETH Zurich, Virtools and Green Dino. Several months were spent on defining proposals for national and European financing. The scientific commission of the faculty of architecture and its dean at that time, Professor Hans Beunderman, who became a supporter of those plans. The dean and professor Oosterhuis convinced the board of the university to create conditions that would allow bringing this ambitious project to life.

As the iWeb was being installed on the TU Delft campus, Hyperbody has started the research for the design environment to be inserted inside the pavilion. The test setup was fit in a 64m^2 room and was given a name Protospace 1.0. It was planned to be smoothly transitioned to its final version: Protospace 2.0 to be located in the iWeb. With university ICT innovation budgets, Hyperbody researchers performed the research of multi-faceted research into design and engineering of such “collaborative design environment”.

The research was multifaceted. One aspect of it was the design of the space of Protospace itself, done by ONL in very close cooperation with Hyperbody researchers. The other aspect was the design and creation of the entire software system that would drive Protospace and introduce new qualities to the ways in which architecture was to be designed. Last, but not least, many technological possibilities had to be investigated in order to make the system operational. The involved technical resources, involved among others the use of three-dimensional stereo projections and development of alternative interfaces with which the multidisciplinary design teams would form virtual designs in Protospace. This research has been taking place on many parallel tracks and was included in several autonomous research projects, all of them supervised by professor Oosterhuis. Since the start the programming of core systems of Protospace was done in a computer game development platform, Virtools Dev, later rebranded to 3DVIA Virtools.

Virtools has been extensively used in Hyperbody for fast prototyping of interactive systems. It became facilitary for programming the Protospace Demo projects. Virtools provides a comprehensive set of reusable classes and methods for building interactive 3d environments, combined with an intuitive GUI (graphical user interface). All classes of objects that can be instantiated as objects in the constructed 3d environments can be extended with scripted behaviours. Through the hierarchy system and group class, virtools objects can be combined into assemblies of objects that can all share one behaviour.

In Virtools all virtual objects can be created as autonomously operating agents. Each such agent can be based on a pseudo-class; a template assembly of native Virtools classes with specific properties and including the scripted behaviour. Such pseudo-class would then be instantiated in the virtual model in one or many virtual agents. During its operation, the agent could change its own properties and interact with other agents.

The basic functionality of Virtools was demonstrated by the virtual model of the shop design by Nelson, where parts of the designed shop were scripted to respond to the user avatar, all of them scripted individually. The Flockject project of ir. Christian Friredrich went one step further. Over hundred virtual objects scripted in Virtools were programmed individually to keep the set distance to their direct neighbour objects and to the user avatar. As a result a “flock-object” emerged out of individual agents, which would always follow the user and locally deform depending on his or her position in relation to the swarm. These and other experimental projects triggered imagination and show possibilities for more complex applications of such distributed systems to architecture and their potential role in the Protospace.

c) Evaluation

The initial development of protospace 1.0 has identified a multitude of promising directions for development of a collaborative design environment for architecture. Perspective of iWeb realization as a laboratory for hosting such environment has provided a tangible goal. However, initially conducted experiments lacked coherence and shared plan. For this reason, Protospace Demo projects were consequently initiated, with the main goal to provide a coherent design environment that could be used in the iWeb pavilion upon its realisation.

2.2. Protospace 1+

a) Outset

To crystallize all concurrent investigations on Protospace, in 2004 it was decided that the best way to progress on the work is to design the eventual Protospace system by developing them throughout a number of operational experiments. Initiated then “Protospace Demo” projects[1] were aimed at realizing the Protospace in a series of experimental design environment setups in the Protospace 1.0 environment. The aim of these projects was to make Protospace system ready to be installed directly after the realization of the iWeb.

The first Protospace demo 1.1 project started with an intention of building a quick, working prototype including all fundamental functionalities that Protospace was planned to include. The author, at that time a student assistant at Hyperbody, was given the task to develop the prototype of that system. The content and design of it was agreed upon during several brainstorming sessions led by professor Kas Oosterhuis, with Dieter Vandoren, Hans Hubers and Nimish Biloria joining the group.

The leading challenge for the design environment was to provide each working in it designer with a distinct, role-specific “view” on a designed architectural object. Professor Oosterhuis proposed four hypothetical roles for the team members. A “visionary” was responsible for the holistic qualities of the project and its consistent appeal and operation. A “qualifier”, for assigning materials and form adjustments to individual design elements. A “validator”, for making sure that the project is structurally and physically sound and a “calculator” for checking the feasibility of the project.

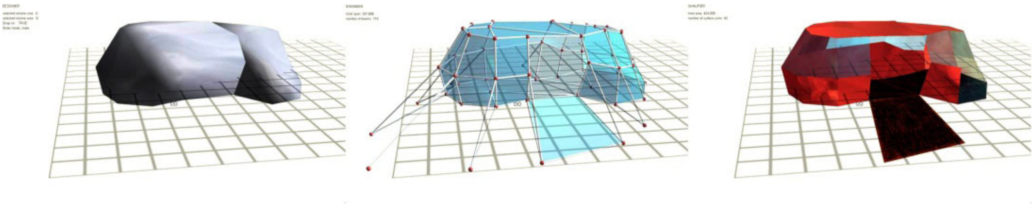
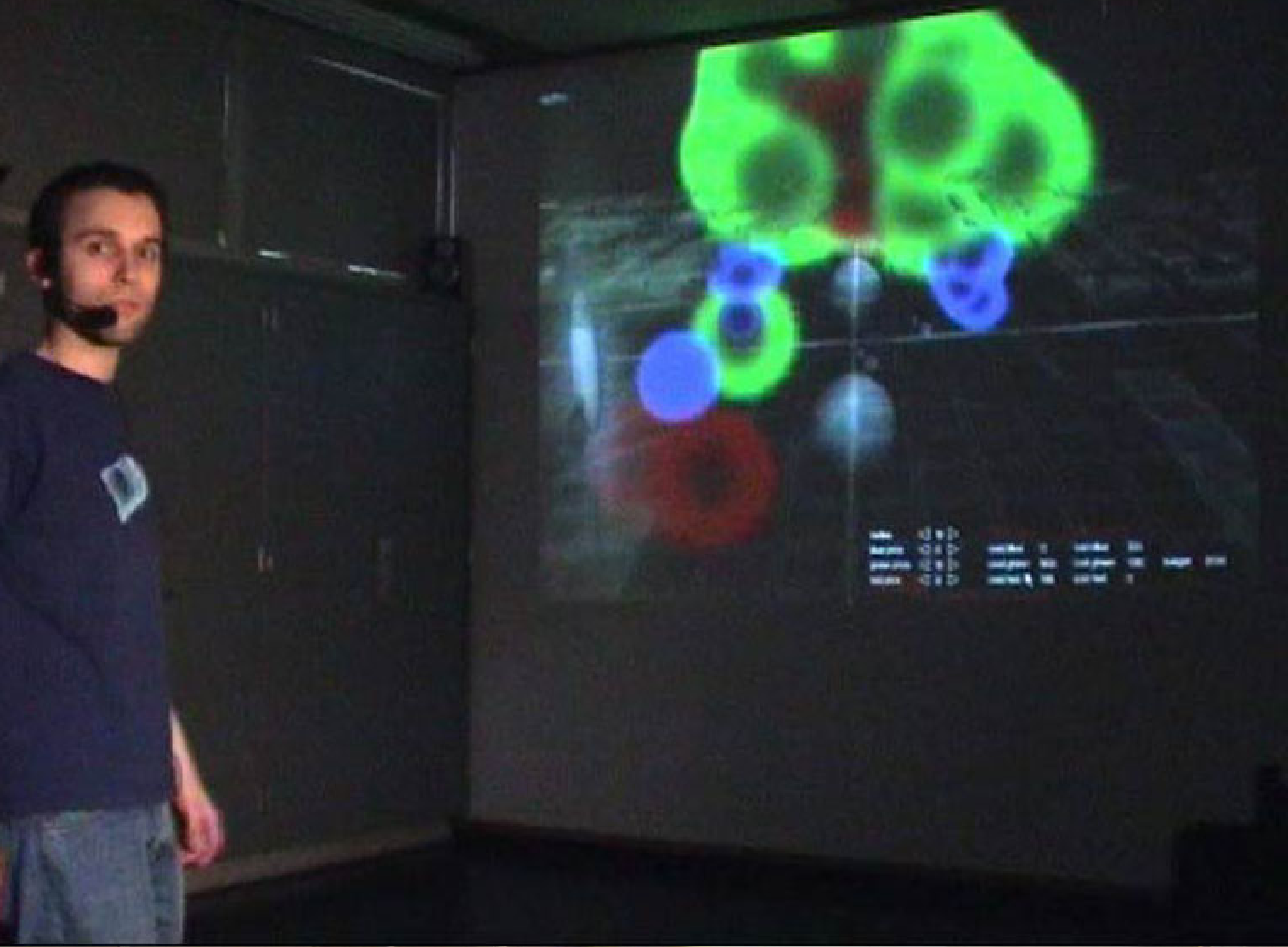


Fig.51. Screenshots of different views on the project generated in real-time in demo 1.1

[1] Bert Bongers, ‘Multimodal Interaction in Protospace 1.0’, in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012), 533–548.



The design application was initially expected to support these four roles concurrently, but due to technical constraints and development time limitations, the eventual prototype was dealing with those tasks in a sequential manner. The functionality was very simplified and not ready for real-world projects, but after a series of use tests, the overall system was positively evaluated.

b) Protospace Demo 1.2

After completion of the Protospace Demo 1.1, it was decided that the development needs to be continued. The critique on the demo 1.1 project was that it has been aiming too much at achieving the functionality which was already to some extent commonly available in other, commercially available programs, such as 3d mesh modelling suites for form modelling or spread sheet applications for cost calculations. The project team concluded that systems provided in the Protospace environment should not mimic features already available on the market, but provide unique and focused solutions, specific to the new ways of designing that Hyperbody has been promoting in its research and education.

In the practice of ONL and many earlier works of Hyperbody, the idea of swarming design components had been applied with well evaluated results and provided novel design opportunities. It has been decided that in addition to the provision of specialized “views” of the design environment (as explored in Demo 1.1), one additional, immersive “view” on the project was needed. Such view would explicitly allow designers to see the design as a network of interactive “swarming” elements. At the same time, it has been indicated that a preferred solution was to use existing software applications as profession-specific, expert “views” on the project, to replace solutions prototyped and tested in demo 1.1. This problem however was not yet to be extensively addressed at that stage.

Additionally application-specific interfaces were built in order to allow designers to interact with the deployed agents. In case of Demo 1.1, there were four different interfaces for four experts. Each interface would additionally hide the agents considered irrelevant for the given expert. In this way expert “views” were created. In Demo 1.2 the interface was to be shared by all members of the team.

With the special focus on development of interfaces, dr. Bert Bongers joined the team of Protospace Demo 1.2. The important feature of the application being developed was that it had to allow its users to intuitively interact with its content. Dr. Bonger’s extensive knowledge and experience with interaction and sensor techniques provided needed support in development of interfaces for the environment.

Consequently, after four months of development, a test installation was set up. In that installation the design content was represented as a cloud of multi-coloured particles. The particles were capable of repositioning themselves, based on the proximity to their neighbours and locations, and parameters of attractors placed in the design space. Four design team members could control this content. Two of them were able to add, delete and change properties of the particles of space. Their virtual actions were controlled by wireless joysticks and tracking of their selected movements. The fourth person in the team had a more passive role. Its position in the design room was directly translated to the position of a virtual avatar in the design space and all data surrounding it was collected and displayed on screen. In this way the fourth team member could with his or her position in the space of the design room adjust localized parameters in the virtual space.

c) Protospace Demo 1.3 and 1.4

Demo 1.2 was well received and highly evaluated. It performed as a test bed for numerous design scenarios and it proved to be a successful demonstrator to potential 3rd party collaborators and clients. However, the developed technical solutions were experimental and

Img. 18. Left: a test session in protoSPACE 1.1

in some cases done ad-hoc. The system was not robust enough to operate in a real-world scenario. In demo 1.3 the same team had decided to update the interfaces of the system, by validating many possible controllers for the application. At the same time the software prototype was being improved and applied to two different real-life projects on urban design scale, as part of the Protospace demo 1.4, in order to validate its practical usability. The first one of them was located on the site of Technopolis, adjacent to the campus of TU Delft. The other one was the Manhal Oasis project developed at ONL [Oosterhuis_Lénárd], as earlier described from the methodological perspective. In both cases the application's shortcomings were systematically identified and improved as its development gradually progressed.

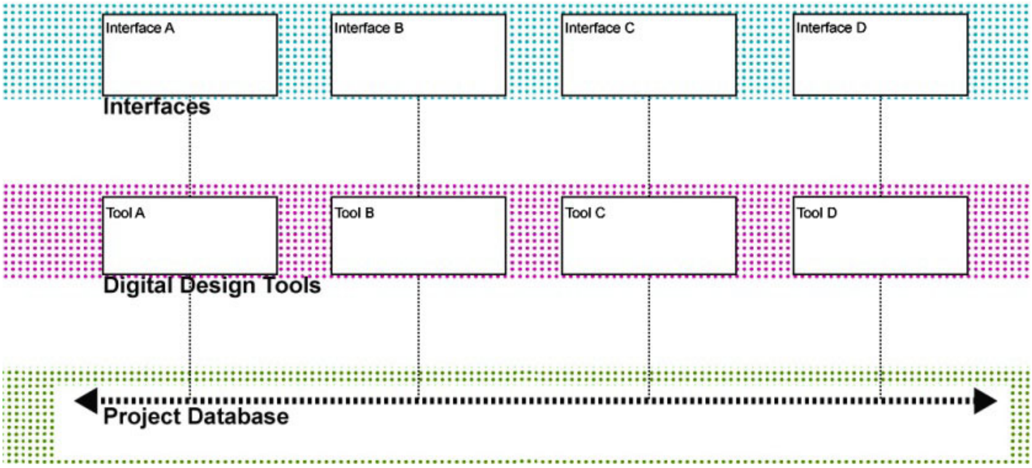


Fig.52. Structure of the design environment introduced in Protospace demo 1.4

At this stage, first attempts were also made to connect different commercial software applications to the demo 1.4 system. Simple database was used for making this real-time data connection. Interfaces developed in demo 1.3 were connected directly to that system. While testing these solutions it has been observed that despite its usability, this setup still had several shortcomings. The shortcoming were coming from the lacks in system flexibility. In real-world architectural practice, design challenges vary. As a result every project requires a different set of functionalities from the design environment, hence also different software can be employed and virtually created components may have different roles across projects.

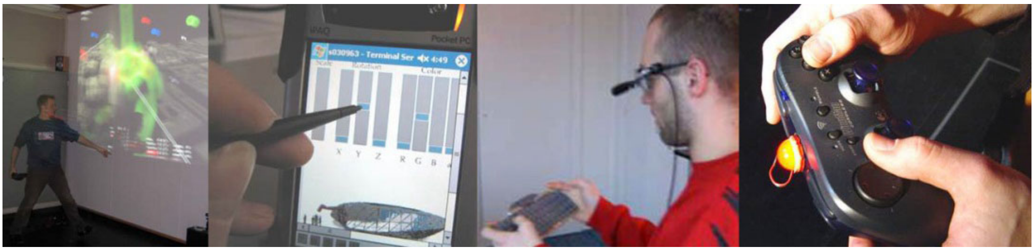


Fig.53. Selection of user interfaces applied and tested in Protospace demo 1.3 and 1.4

d) Evaluation

The iWeb building has been completed in 2007. All equipment and hardware has been installed inside and Protospace 1.x Demo developments were approaching the 2.0 version. The developments of the collaborative design environment have progressed throughout the Demo projects and a wide range of solutions has been delivered. However, numerous problems have also been identified. Constructed prototypes showed the wide range of possibilities, but they lacked the functional polish and user-friendliness. Setting up the system had been a time consuming activity and errors were frequent. Above all, the tested solutions only dealt with selected aspects of executed projects, while the aim for Protospace 2.0 has been set to deal with projects comprehensively, involving the entire design team. Therefore further practice oriented developments of the Protospace system were scheduled to take place in the new laboratory location.

2.3. Protospace 2.0

a) Outset

The development of Protospace 2.0 in the iWeb pavilion was two-fold. The first aspect involved the setup of facilities in conjunction with the spatial organisation and modes of operation of the laboratory. The second aspect involved bringing the design environment developed in Protospace Demo projects to practical application through comprehensive case studies[1].

b) Setup

The Protospace 2.0 has been designed as a physical environment serving as an interface to the virtual design environment for a team of experts. It has been made to facilitate team collaborations in a variety of ways. The central part of the laboratory was five translucent back-projection screens hanging in a pentagon. Screens were not touching each other, so that access to the inner part was possible at any time. Various screen configurations were possible by lowering and raising the screens.

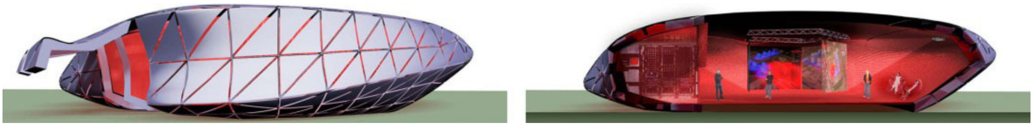


Fig.54. Protospace 2.0 interior

This setup allowed dynamic collaboration sessions. In those sessions up to five team members or expert groups could use the screens individually by standing on the outside. In that setup each screen would deliver a different view on the system, specific for the expert's role in the team. Experts could subsequently come all together and evaluate the integration of their individual work while a shared view would be inversely displayed on the five screens around them in an immersive way. In between semi-configuration were also possible. For example three experts would use the individualised work spots, while two others will discuss specific aspects of project integration in the middle using only two of the five screens.

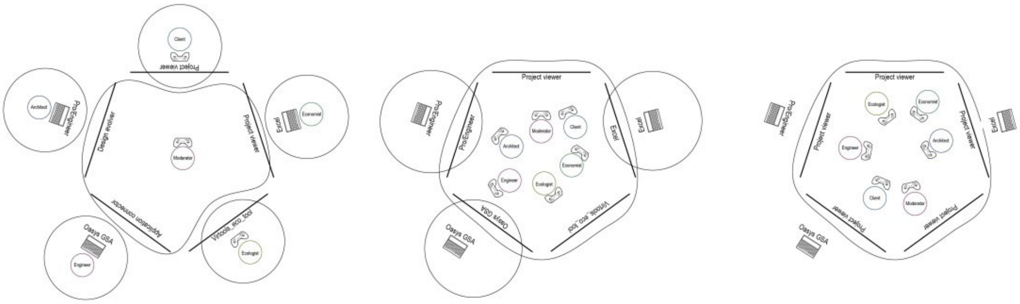


Fig.55. Various moments in the design process at Protospace 2.0

[1] Kas Oosterhuis et al., 'Protospace 2.0', in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012), 549–556.

Distribution variations - organized
@ Manhal UAE
Distribution variations - organized
Programme distribution FAR 4.0
DOE HET ZELF
MEINEMA

The work would be done mostly standing, as such is known to increase efficiency of decision making and collaboration. For this a number of handheld interfaces were provided, including gyroscopic pointers and Nintendo wii controllers, PDA/smartphone based dynamic touch interfaces, wireless gamepads and handheld keyboards. Each screen had its own server computer, where all computers would also have a cluster operation mode for synchronised display of immersive environments.

Protospace 2.0 in the iWeb could be described as a meta-instrument. It provided an intentional spatial organisation and facilities, while not super-imposing and specific software solutions or workflow, allowing a number of specific design instruments to be used as design environments and validated in the process.

c) Environment development.

Protospace Demo projects have resulted in the general structure of Protospace systems. However, further research needed to be carried on in order to test and develop further the actual functionality of the design environment. Following the opening of the Protospace 2.0 laboratory, two case study projects were initiated, the floating city and the A2 city. Both projects used Virtools as the core environment for virtually deploying a system of autonomous design components in a collaborative design scenario.

Chris Kievid, Dieter Vandoren, Tim Mcginley, under supervision of prof. Kas Oosterhuis and in cooperation with ONL have developed a working prototypical Protospace project titled Floating City. This project was a dynamic parametric model, which allowed different specialists to change its parameters in profession specific “views”. The Protospace interfaces have been applied to directly control parameters of modelled objects. The case study project focused on providing multiple design team member one, comprehensive, immersive view of the designed urban-scale environment.

In parallel Christian Friedrich and the author developed the system for the A2 design studio. Here the focus was on parallel design. The group of Hyperbody Msc1 students was given an assignment to design the urban environment along the Dutch A2 highway. Each student was responsible for one sector along the highway, being required to relate to adjacent sectors of other students. Design environment was built on the Protospace Demo 1.4 foundation and was extended with an online front-end. This meant that not only five Protospace screens could be used for displaying different sectors of the project, but also designers could be remotely connected over the internet and simultaneously contribute to project development.

Additionally, the system included another view, which was overlooking the entire development in a top-down manner, establishing global parameters, accessible from all different zones. These parameters were: overall program values for specific functions to be distributed throughout the whole design site and included guidelines for building heights along the highway represented by a three dimensional NURBS curve. In this way each point on the ground plane in the entire site was mapped to a specific preferred height value. However it was up to individual designers whether to respect or override this parameter.

In order to simulate program demand distribution occurring in real life economy, insertion of each element of the program was causing an additional demand of other functions to appear, this demand was spreading to adjacent zones, while its value would also exponentially decrease. A special matrix model was used to calculate demands for different functions.

d) Termination of Protospace 2.0 and evaluation

The further development and testing of Protospace 2.0 laboratory was prematurely terminated by the fire of the faculty of Architecture in May 2008. Due to losses in technical equipment, the closing of the facility and partial losses of data and developed software, the systems could not have been developed further as planned.

Img. 19. Left: work sessions in protoSPACE 2.0

2.4. Discussion

The unfinished development of Protospace 1.0-2.0 design environments has provided significant research material for further development of design systems. Numerous case study applications have proven the applicability of developed instruments to iA design challenges. Parallel collaboration of design team experts has been significantly facilitated by the development of custom "views" on the virtual prototype. However, it has also been concluded that in-house development of such views is unfeasible and to a large extent unnecessary as existing commercial software can replace specialised views and directly connect to shared project database.

What has proven to be the main bottleneck has been the centralised approach to the project data and lack of shared conventions and protocols. XiGraph initiative has provided the initial roadmap for realisation of such protocols.

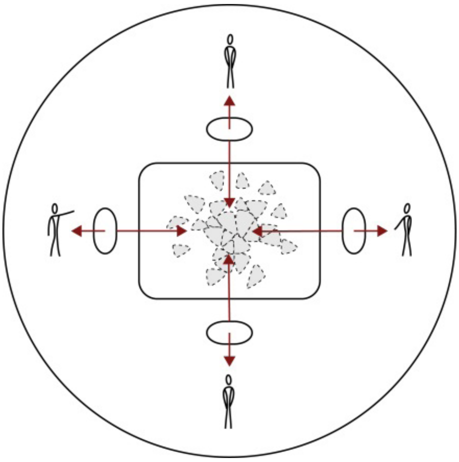


Fig.56. Ideogram: Shared virtual design and prototyping environment

Conclusions summary:

- A building system developed in a virtual reality environment can be used as an early design stage prototype.
- Shared virtual model enforces design parallelisation and fast design iterations
- Developing a fully customised virtual environment for architectural project development is an unfeasible task.
- A shared virtual design environment faces problems of scalability and extensibility.

3. protoKIT – evolving ecosystem of design instruments

Challenges:

- Integrate design platforms commonly used in the community.
- Distribute the virtual design environment.
- Increase non-expert access to design instruments.
- Introduce a protocol for exchange of information and interaction between instruments.

The instruments developed up till the termination of protoSPACE 2.0 laboratory have shown several shortcomings. The integration of external expert software has only been prototyped and not integrated in the workflow. The design environment has been centrally structured, not allowing branching or integrating of projects. Participation of non-expert users has not been explored. Most importantly though, no explicit protocols or conventions were established to ensure interoperability of developed design instruments and their versions. As a result, continuous improvement of prototypes and their integration has been troublesome and in many cases developing new applications from scratch turned out to be easier than continuation of the development of old ones.

Reflection on the terminated development of protoSPACE has raised a number of questions about further development of the collaborative design systems. In discussions among Hyperbody researchers, it was pointed out that a laboratory is not directly essential for existence of a virtual design environment. Instead it can be considered as an extension of such environment, allowing use of specialised interfaces, holding collaborative design sessions or facilitating virtual and actual prototyping of developed architectural systems.

As presented to this point, there had not been one consistent application, software or system developed for protoSPACE. protoSPACE from version 1.0 to 2.0 has been developed on several parallel trajectories, which influenced each other and which were combined in different configurations for different case study research projects[1]. After the shutdown of the protoSPACE 2.0, the design instruments developed for the laboratory have lost their originally intended context. However, many of them, although originally developed for design sessions in protoSPACE, became also usable in other design situations. Some of them had been early-on employed in projects such as SNPC or Manhal Oasis of ONL[Oosterhuis_Lénárd] or author's own Paracity project, as well as in many other projects by other members of Hyperbody.

Following this turn of events the name "protoKIT" has been retrospectively proposed as an umbrella label for continuation of the development of protoSPACE 2.0 systems beyond the confines of the no longer operational protoSPACE 2.0 laboratory, and perversely embracing the later developed protoSPACE 3.0 laboratory as one of the protoKIT instruments rather than a container for a complete set of protoKIT instruments.

protoKIT has not been a thoroughly planned initiative. Instead, it has emerged out of a number of individual initiatives of Hyperbody researchers, which eventually converged in one meta-system of diverse kinds of architectural design instruments. protoKIT can be in retrospect defined as an open and extensible "kit" of design instruments, targeted at development of architecture as a multi-agent system and bridging the virtual and the physical dimension of architecture. It may contain any kind of an instrument serving this statutory purpose, as long as this instrument connects to other instruments of protoKIT. Such instruments can

[1] Kas Oosterhuis et al., 'Protospace Software' (presented at the Second International Conference World of Construction Project Management, Delft, the Netherlands: TU Delft, 2007).

thus include a) software applications; whether purposefully developed as parts of protoKIT, or commercially available software applications extended with plugins for connectivity with other protoKIT instruments b) screen-based, tangible, gesture, voice and other interfaces between humans and the virtual environments c) interfaces between software and specific virtual agents, including protocols for data exchange, database and BIM systems and others e) interfaces between designers f) material prototyping platforms, including fabrication and embedded systems. Yet, the list is only indicative, as other types of instruments could potentially also be added.

3.1. protoBASE

a) Outset

The fundamental problem when working with multiple design instruments is how to store and share project data between them? In static systems the speed of data access and exchange is not critical to system operation. However, when the system agents need to communicate rapidly while being distributed in various software platforms and in the physical world, the way in which such communication is organised becomes critical to the functionality of the system.

b) Development

Throughout the Protospace research a number of solutions were used to store and share data. Initially the fastest ad-hoc methods were chosen. Later more refined options were employed.

Data files

Data files are the most common way to store and exchange digitally created project data. In Protospace Demo 1.1, a simple, self-made, calculation sheet was embedded. This has shown how futile it may be to attempt to reinvent software functionality which has been already perfected over decades of development, such as spreadsheet software. Consequently, Microsoft Excel has been used in later instrument sets for performing all operations where continuous overviews of large amounts of data were needed and where quick calculations had to be performed.

Interoperation between protoSWARM and MS Excel was initially achieved through exporting all relevant data from Virtools arrays as tab-delimited text files, importing them in excel, performing calculations, re-saving the file and importing it back in protoSWARM. This allowed for very reliable operation; however the data could not have been adjusted in real time. In projects like Technopolis this resulted certain inflexibility in the way in which global variables, derived from e.g. cost estimates and feasibility studies were dealt with. Since a relatively time consuming action of importing the file, editing and saving was required to change these numbers, they were edited less frequently than parameters directly adjustable in the protoSWARM environment. As a result, if smoother integration with the spreadsheet was possible, significantly more project scenarios could have been explored.

MySQL

Although spreadsheet software is suitable for ad-hoc overviews and calculations in medium sized data sets, it cannot be used reliably for storing or calculating large amount of data. Additionally, connections established to MS Excel had many limitations, the data had to be exchanged through a text file, or, in Christian Friedrich’s SDK connection solution, the spread sheet had to be open on the same machine as the other connected to it applications.

Usage of a database has been an alternative solution for both data exchange and storage. Already in Protospace Demo 1.3 a Microsoft Access database was used to store project data. Speed tests were made, and single record refreshing could have been achieved multiple times per second, even if the database was running on a remote server. From a MS Access database, a separate connections was made to an Excel spread sheet allowing Excel to automatically extract needed data sets and save others back into indicated records. Eventually similar setup was achieved with a MySQL database in place of Microsoft Access, which allowed for more possibilities in usage of stored data in dynamic online content. Online database access has provided an alternative for a spread sheet scenario, for situations where data needs to be accessed through a computer console without any specific preinstalled spread sheet software.

The storing of data on a database, to which a number of applications could have access provides a flexible solution for near real-time data exchange, combined with data storage. Two common approaches to storing project data exist. One approach stores the history of project creation, as a sequence of operations and parameters (e.g. Generative Components, COLAB). The advantage of such approach is the ability to trace the history of creation of a given output and re-enact that history when retrieving a project from its previous, saved state. Such approach is, however, not applicable to largely complex and distributed systems, where emergence of given outputs is dependent on local interactions and feedback loops. In that case only the storing of a system state is reliable, which is the most common approach. Saving of system state can be extended with selective history, where, e.g. individual agents could “remember” the history of their own development in case that would be considered relevant in the project.

In demo 1.3 data was stored in database records, where each record would correspond to one agent and a database table would correspond to agent types. This approach proved intuitive in applications.

However, with increasing number of connections between applications, the management of these connections became problematic. Each process of storing and retrieving information had been pre-programmed in the protoSwarm, MS Excel or other application. What was repeatedly encountered in design case studies where the system was applied, was that connections between different parameters, often originating from different applications were formed ad-hoc and were difficult to predict. For example, tessellation density of a building façade dictating the layout of façade components, could be changed from a globally defined static variable to the locally determined estimate number of passers-by in front of it, or the operation of a dynamic shading system could be remapped from responding to sun angle, to the commands sent from individual users and neighbouring components (in case of the Muscle Façade project). Such alterations in the dependencies between variables were performed continuously in developed design systems. The lack of ability to flexibly and rapidly perform such changes became a bottleneck in design processes and was identified by designers as the limitation of their ability to creatively explore design alternatives and to test their performance.

XiGraph

The XiGraph was introduced by Christian Friedrich as a remedy to the aforementioned situation, building upon selected ideas, defined earlier in his BehaviorLinks prototype. As its author describes “XiGraph is a metasoftware system for connecting applications and other datasources in real-time as well as a basis for building parametric modelling systems. In this way it facilitates creating a real-time network of design and construction operations in applications and hardware devices. XiGraph will be developed and implemented as a software library and applied in the prototypes and in Protospace. (...) XiGraph, is generically a semantic network. Semantic networks can be processed by computers, and be read by humans with relative ease. Any structure of data in a computer can be represented within a semantic network. In this way, XiGraph system provides a generic method for describing data structures and their content within the same data encoding specification. The XiGraph description can be saved and communicated in an open format (e.g. XML). It can then become a unified mode of communication between the different elements of the Protospace system architecture, different applications, hard drives, networks and hardware components like sensors and actuators. Next to giving adequate representation of the data structures within the software and hardware components which it connects, XiGraph can also represent the connections themselves. The non-hierarchical nature of XiGraph allows for adaptive computational design approaches which are dependent on feedback loops, like cellular automata, neural networks, swarm modellers and the like. The generic nature of XiGraph provides many possibilities for implementation, and ensures compatibility with future developments."[1]

protoBIM and quantumBIM

The idea of protoBIM and quantumBIM have been introduced by prof. Kas Oosterhuis as convergence of the solutions developed and discussed to this point with current industry standards, providing an outlook towards the future development and broader adoption of discussed direction. The term BIM means either building information modelling or building information model. In the first meaning BIM refers to an integrated design process in which one complex parametric model is constructed, in the second meaning it means the model created in that process.[2] In software such as Autodesk Revit or Gehry Technologies Digital Project proprietary building information models are used for exchange of date within that software. Industry Foundation Classes were independently developed to define standard exchange data format. Projects such as bimserver.org, are attempts to provide data exchange repositories that could connect building information models across different software platforms.

However, the very notion of "model" in the context of BIM becomes devaluated in the context of presented approach. When buildings are being developed as systems, there is no more need for representation. Instead of a model, there is a prototype, an always operational, actual building system. The BIM technologies, however, can very well be adapted to the complex adaptive system prototyping. For this the name protoBIM has been introduced and used. As prof. Oosterhuis describes the concept, protoBIM "supports development from a written conceptual statement via a swarming behavioural point cloud towards a BIM that contains all required data for building approval and the tender process. protoBIM connects all relevant disciplines to each other". Such protoBIM, however, aims below the ambitions that the earlier explored techniques for data storage and exchange (MySQL, XiGraph) approached. protoBIM only applies to a virtual prototype of the building system, while the aim of MySQL and XiGraph combination was to provide a data exchange system, that would also apply to embedded architectural components operating in the physical space. Such extension of the protoBIM, being focused on real-time data exchange has been labelled quantumBIM. "The leap from protoBIM to quantumBIM must be made when the design subject is a programmable structure, where the support for streaming data is mandatory." Quantum BIM is "parametric in its nature, open to imposed streaming external data, deducing its internal consistency from the bi-directional relations between the actors of the point hive. Based on

[1] Ibid.

[2] Chuck Eastman et al., *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors* (Wiley, 2008).

the continuous transaction between the points, that hive is in a state of continuous evolution, building up the critical mass needed to eventually make substantial jumps in its evolutionary development."[1]

quantumBIM can be considered to be one of the agents in a developed architectural system. The role of this agent is to facilitate exchange of information between agents and to collect and store data from them. It is not a compulsory agent for building operation, as agents can also communicate locally among each other, but it permits connection of agents of different types, that operate in different mediums.

c) Evaluation

The manner in which data storage and exchange among instruments has been dealt with has gradually evolved throughout the design experiments. The path can be summarized as first increasing the speed and access to shared information, further providing a shared set of conventions for structuring and understanding of that information and eventually distributing and adding flexibility to previously centralised design information repositories.

3.2. protoSWARM

a) Outset

Of the Protospace 1.0-2.0 instruments, the most widely used was the multi-agent design environment "protoSWARM", which for a long time had been developed without being given an explicit name and has often been referred to as the "swarm tool". Its first version was developed by the author for the Protospace Demo 1.2 project. Later its variants were employed in many of the projects described in section 1 (including, SNCP, Automotive City, Manhal Oasis, Paracity, Muscle Façade, Leafs portal and others). The premise of protoSWARM has been to deliver an open, extensible and easy to use virtual environment, in which designers with diverse expertise could prototype autonomous virtual agents corresponding to the design components they are dealing with, and consequently develop and virtually deploy complex multi-agent systems as aggregations of these agents.

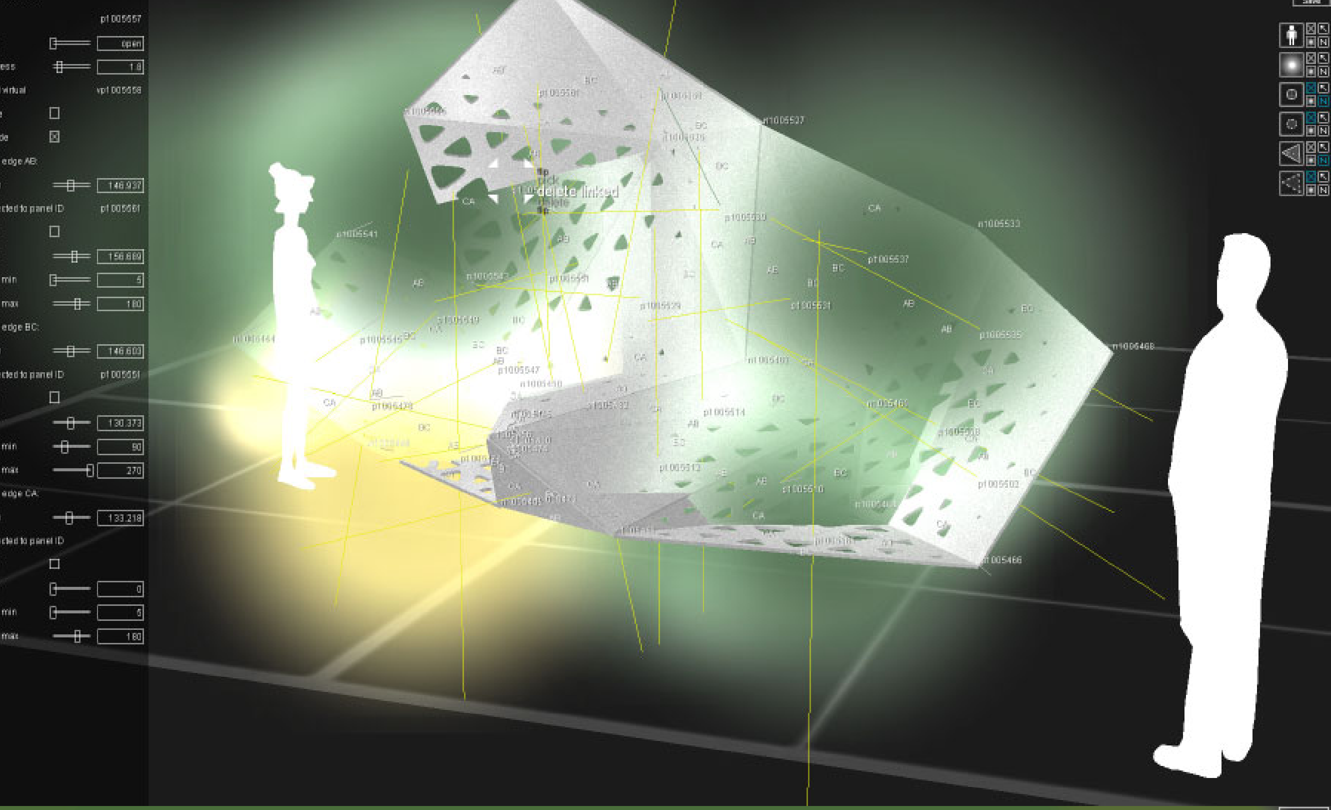
b) Development

Initially three kinds of agents were used in protoSWARM, namely: functional program particles, attractors and relations. Its initial goal was to accelerate and empower the process of architectural programming, using an approach independently investigated by Ophir[2]. The functional program particles were initially very simple agents. Their parameters included position in local Cartesian coordinates, floorspace in square meters and function type. The graphical representation given to functional program particles was a spherical shape with a blurred boundary. The size of that shape would directly correspond to the floorspace value. The area of the section through the centre of the sphere equalled the floorspace. The colour of the particle would correspond to the function type. In the initial version the function types were constrained to four generic categories; commercial, office, housing and leisure, as first applications were on very generic and simplified urban scale.

The attractor agents in protoSWARM initially had two types, represented by volumetric points or lines in space, defined by respectively one or two local Cartesian coordinates, range, and strength of influence. The intention for attractors was relay to the system all kinds of external

[1] Oosterhuis, *Towards a New Kind of Building*.

[2] Yaniv Ophir, 'Collective Intelligence: An Agent-Based Approach to Programmatic Organization in Architecture', *International Journal of Architectural Computing* 7, no. 3 (2009): 479–499.



pt 005657

open

ess 1.8

virtual vp1 005658

□

de ☒

edge AB:

146.937

cted to panel ID pt 005661

□

156.889

min 5

max 180

edge BC:

146.603

cted to panel ID pt 005651

□

130.373

min 90

max 270

edge CA:

133.218

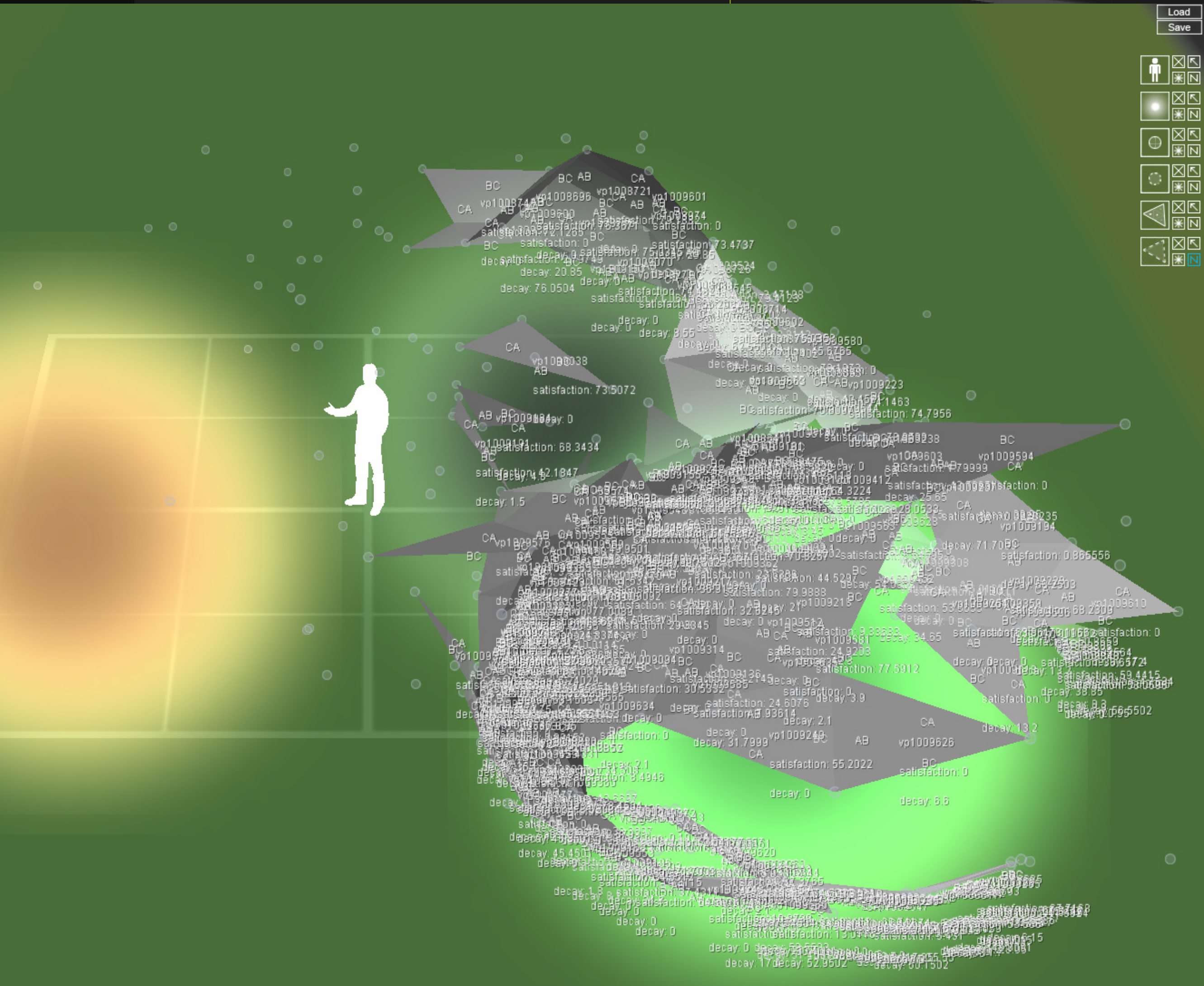
cted to panel ID

□

0

min 5

max 180



influences and constraints, as well as provide a way for designers to influence the particle swarm without directly controlling any of its particles. If a particle was located within the range of influence of the attractor, the attractor would move it closer or further away in fractional steps, depending on the sign and magnitude of the strength value.

The relation agents were always connecting two of the functional program particles. Initially relations could only affect the relative distance between particles. Their two parameters were distance and strength. Once the distance between particles exceeded the specified value the parameters would be moved towards each other, affected by the strength parameter in a similar fashion as in case of attractors.

The interface of the system was simple. A pointer was used to select an agent and once selected, a set of agent's parameters was shown on the bottom of the screen. Consequently a designer could alter these parameters. Additionally designer could delete the agent, split it into two, or merge it with its closest neighbour. There was also an option to move the selected agent or create a new one. In the Demo 1.2 setup, up to four designers could work simultaneously. The 3D environment was displayed on two adjacent screens and was tested in setups with up to five screens. The pointers of individual designers were simultaneously operated using wireless joysticks. In following conducted design experiments abilities of individual designers were constrained. One of them could only interact with attractors; the others could either move the functional program particles, change their parameters, or delete, create, join and merge them. In the application a number of urban plan development scenarios were tested by Hyperbody staff and students taking the roles of individual designers. In all cases designers could explore a wide range of program distribution alternatives in a fraction of the time that it would take using traditional techniques. Constraining individual responsibilities of designers reduced the amount of conflicts.

Following the Demo 1.2 project, the protoSWARM was modified to a screen-mouse single user interface. Consequently it was applied in a number of projects. For each of these projects the system was heavily modified. For SNPC, an unlimited number of functional program types were added to the particle agents, distance relation parameters also included a threshold, with a minimum and maximum distance value. This allowed for the design process described in section 1.1 to take place.

Many more features were added for Speed and Friction projects. There the particles were replaced with volumetric cylinders and the agents received an additional parameter of height and stretch, consequently resulting in a direction vector. Their behaviour also became enriched, depending on the setting of the stackability flag, functional cells would stack on top of each other if displaced or remain on the current level, pushing other cells away. A third type of attractors was added, a curve, corresponding to design powerlines and transportation routes. Function cells would in such case get repelled or attracted to and from the nearest point on the curve. The attraction or relation would now also affect orientation of cells if selected. This made it possible to develop the streamlined organisation of the entire project.

For Manhal Oasis, a different set of features was introduced. There, the scale of the project was larger and consequently functional program cells additionally had a parameter of the number of floors. Most significantly, however, a new type of agents was introduced: global agents. Global agents had no embodiment, or in other words, represented aspects of the project as a whole. Global agents summed all extrinsic properties of belonging to them elements. For every named category of the functional programme a global agent existed. One meta-global agent existed for all global agents. Every regular an global agent was consequently equipped with a Boolean flag called fix parameters. If a global agent had fixed parameters, change of parameters in its sub agents would result in balancing the parameters among agents themselves so that the global value would be maintained. On the other hand changing

Img. 20. Left: protoSWARM employed in the d|e|form project the global value would only affect local agents with the fix parameters flag unchecked. In this way, the earlier described complex development process of the Manhal Oasis masterplan could be achieved.

A new set of agents was developed for applications such as Muscle Space, Muscle Façade or leaf portal. There an entirely new set of agents was added to the system, where each agent type was corresponding to a specific building component. Components such as solid struts, flexible struts, fluidic muscles, fixed joints, flexible joints and surfaces were developed. Each of these virtual agents had properties corresponding to properties of physical components. The virtual components were not attempting to accurately simulate the behaviour of their physical correspondents. They did, however, include the features that were critical to develop similar holistic behaviour in virtual and corresponding physical systems. This allowed testing of numerous behaviour scenarios that would otherwise be too laborious to prototype. Eventually the physical behaviour of realised prototypes in all aforementioned projects was controlled directly by virtual agents, since no technical resources were available to embed the physical components with embedded computing devices. In this way, the distributed behaviour of system agents occurred entirely virtually and through serial and usb connections all sensors and actuators were mapped directly to individual virtual agents and acted as their extension. In this way hybrid operation between the virtual protoSWARM and the physical world could further be developed and tested directly with human users.

Despite its broad applications, work with protoSWARM became difficult. Its different versions with varying functionality were not compatible with each other. Gradual improvements to different layers of the system were made independently on different versions. Gradually, due to different demands, conventions for data exchange between agents changed. In 2008, 6 different interface versions were in parallel use.

Additionally, gradual changes to the application and ad-hoc improvements made it illegible and difficult to further modify. Due to the complicatedness of the scripts and the structure in which they were set to operate, it also was not feasible for other designers, not only students, but also other experienced in Virtools programming Hyperbody researchers to add new agent types or add any other modifications.

Consequently it became clear that a new approach needs to be taken. After the faculty fire, the author has rewritten the protoSWARM from ground up in a modular manner. The modules of protoSWARM interface, modules of individual agent types, modules for data storage, exchange and connections to the physical devices were separated. A general convention for exchange of data among agents, with the interface and other modules has been established. This has allowed further development of the design framework and easy integration with other protoKIT modules.

c) Evaluation

protoSWARM has shown to be an essential instrument for design of complex adaptive architectural systems. In projects to which it has been applied it served as an incubator of the designed system, allowing its rapid deployment and testing. Further on, it has also shown the capacity of integration with the physically deployed system.

What has lacked in the instrument is a greater diversity of components that could be used at the outset of each project and could eventually be co-developed by a community of designers, significantly improving the speed and capabilities of new systems,

Most importantly, however, working with Vitrools as the engine of the platform has blocked further development of the platform due to discontinuation of the engine development and to high costs of its licence, required by anyone willing to add low-level content to the protoSWARM platform. Consequently, protoSWARM or a similar instrument should be re-developed as a more open platform around which the iA designer community could further grow.

3.3. protoSHAPE

a) Outset

In analysed design case study projects two distinct approaches were taken in development of the building shape. In some projects, such as Cockpit and Sound Barrier, or protoSPACE 4.0 pavilion, the building shape was designed as one, global agent following other agents such as powerlines. Building component agents would then populate such top-down defined surface form. In projects of greater complexity, such as 751 city or the distributed faculty, where form topology could not have been predefined, the building form would emerge out of local interactions among its components and other system agents.

In all cases the forms of individual building components had to be created with high degree of precision and flexibly, so that individual component parameters could be translated to changes in component's geometric properties. The technique used for this was either parametric geometry modelling.

Traditional 3d modelling software, such as Autodesk 3d Studio MAX or Autodesk Maya, or CAD software such as Autodesk Autocad, allows only limited changes to the once defined shape. In order to provide digital geometries that can be adjusted and adapted throughout the design process, in the Cokpit and Sound Barrier projects geometry scripting, also referred to as procedural modelling, was used. This means that instead of modelling a particular 3d geometry of a building element, the procedure of that component formation would be written as a set of instructions and parameters. For every change to the component, or for creation of many components, the set of instructions can be re-executed with a different set of parameters, and consequently generate different geometries. Such task was however highly laborious, as the scripting of a form can be time consuming and lacks immediate feedback. Nevertheless, procedural geometry scripting approach remains the choice of many architectural designers. Among them, Hyperbody PhD candidate Jelle Feringa co-developed a Python language based geometry programming environment, based on Open-Cascade libraries and entirely devout of any graphical user interface.

b) Development

o Solid Modelling

Solid modelling programs, such as TOP Solid, or Pro|Engineer or CATIA were among the first to allow parametric modelling, where the geometry would be defined based on variable parameters inherently in the program's logic, however the complicated system of setting up parametric relationships and difficulty in its modification was a bottleneck in design processes. Pro|Engineer has been extensively used in projects at ONL[Oosterhuis_Lénárd], while TOP Solid was applied by some Hyperbody students, e.g. in the distributed faculty project.

o Gehry Technologies Digital Project, Atodesk Revit

Another group of parametric architectural modelling software are often referred to as building information modellers. Examples can include the Autodesk Revit or Gehry Technologies Digital Project. The precursors of such design environments initially included limited libraries of components, including only standardised building elements(e.g. Autodesk Architectural Desktop, or ArchiCAD). This has greatly limited the application of these environments to innovative design cases. However, Revit, as much as Digital Project permits easy creation of unique elements. These systems have been extensively used at ONL[Oosterhuis_Lénárd] to create plans for buildings, however their application to dynamic building systems is very limited.

- Generative Components

The Bentley Generative Components (GC) extension to the Microstation CAD platform was first introduced in 2003 and commercially released in 2005 and at a time provided revolutionary solutions to parametric 3d modelling, allowing intuitive scripting and clear overview of model parameters and a possibility to move across the parameter hierarchy tree. GC was extensively used in hyperbody education and some modules e.g. of 751 project were created in that software. What is noteworthy Generative Components has also been the main platform of choice of the emerging architectural parametric design community, organised around the SmartGeometry group.

- Rhino and Grasshopper

As of 2011, the McNeerl Rhinoceros 3d suite with Grasshopper extension, became the strong competitor of Generative Components and has dominated the architectural parametric modelling community. The Grasshopper plugin allows easy parametric modelling using a visual interface. Operations constructed in grasshopper are immediately executed in the Rhinoceros geometry modeller. The open nature of Rhinoceros and its low license price have also led to numerous plugins being developed by the large community. Consequently in 2010 Rhinoceros has become the most commonly used geometry modelling tool for both ONL and Hyperbody projects.

c) Evaluation

In iA projects the role of design of geometry is different than in traditional static and even more so, based on standardized components architecture. Design process involves continuous transformation of designed geometries. Investigated design methods additionally involve continuous reconfiguration of system components. In that respect most solid modelling software and architecture-specific software has proven too rigid for required uses. Ultimately Rhino 3d has been selected due to its high flexibility and ability to extend the core software with functionality specific to encountered problems. It is likely that in the coming years new software suites will appear that will provide a replacement. What is critical in this development is the ability of software to comprehensively model constraints and allow narrowing down of design solutions[1].

3.4. protoSIM

a) Outset

Traditionally, simulation can be defined as a model of reality, which is built to mimic a specific aspect of a corresponding to it real world phenomenon. In the context of protoKIT, and more broadly in the context of complex multi-agent system approach to architecture, simulations are needed to support transition or extension of agents from virtual to the physical reality. However, the ontological position of simulations shifts in these experiments. Rather than being a model of a real phenomenon, simulations were simplified “placeholders”; simplified placeholders of these phenomena in the virtual systems. In that way simulations were aimed to gradually increase in detail and become the physical systems they initially were approximating, without a clear separation between the two. It can be thus generalised that

[1] Axel Kilian, *Design Exploration Through Bidirectional Modeling of Constraints* (MIT Press, 2006).

in protoKIT simulations are not considered to be representations of reality, but are a type of instruments that permit virtual approximations of physical systems' performance. protoSIM is thus a group of instruments used in protoKIT to provide such approximation.

In most tested projects simulations of complex phenomena needed to be carried out before the actual prototyping could start. These simulations included behaviour of energy consumption, wind flows, structural performance, or behaviour of groups of people in and around designed buildings.

b) Development

o Energy

Among employed software applications employed in protoKIT for energy performance estimation, Autodesk Ecotect has been thoroughly tested by ir. Han Feng. In his projects Han employed Ecotect by connecting it directly to a virtual system developed in Rhino/Grasshopper, which allowed him to extend the evaluation of developed by him building system to include the energy calculations, such as energy use, carbon emissions, daylighting and shadowing. In this case the connection was established through a Geco[1] plugin for grasshopper, thus without integration with other protoKIT modules.

o Structural forces

A basic simulation of forces in three dimensional truss structures has been included since early versions of protoSWARM. The simulation has been based on particle spring models (similar in principle to work of Kilian[2]) and allows instant identification of exceeded stresses in structural members. Such approach allowed very flexible validation of structural forces, yet has only been indicative.

On the other hand, in selected projects of Hyperbody connection from multi-agent models in Virtools to structural analysis and engineering applications has been established. Christian Friedrich has performed comprehensive tests between a custom modelling application and the Oasys GSA finite element modelling platform. Oasys GSA permits highly accurate structural calculations, yet requiring significant amount of time to execute.

More advanced and accurate solutions filling the spectrum between the two above solutions have been recently developed for the Rhino platform. Among those Kangaroo[3] and Karamba[4] plugins provide two promising alternatives that are planned to be integrated with other protoKIT instruments.

o Human Behaviour

Behaviour of people is impossible to simulate accurately, as it is governed by individual qualities of humans. However, certain phenomena can be generalised in respect to crowd behaviour, where individualities of humans tend to level each other out in some aspects, such as crowd movement or following the majority. For those phenomena simulations have been custom built in using Virtools or Processing and has been successfully employed in projects such as Paracity or a number of master student projects. These simulations provided valuable insights into projects and showed the potential for such instruments to be employed

[1] Thomas Grabner, 'Geco | Grasshopper', accessed July 20, 2012, http://www.grasshopper3d.com/group/geco.

[2] Axel Kilian and John Frazer, 'Particle-spring Systems for Structural Form Finding', *Journal of International Association for Shell and Spatial Structures* 46, no. 147 (2005).

[3] Daniel Piker, 'Kangaroo Physics | Grasshopper', accessed July 20, 2012, http://www.food4rhino.com/project/kangaroo.

[4] Clemens Preisinger, 'Karamba | Grasshopper', accessed July 20, 2012, http://www.food4rhino.com/project/karamba.

HYPERBODY
PROTOSPA
TU Delft

VOTE AT: HTTP://WWW

in architectural design processes. However, the accuracy of the simulations has not been verified or based on verified research. Therefore these simulations can only be used as an illustration. For real world applications, verified crowd models[1] should be applied and results systematically tested in the architectural context.

c) Evaluation

The iterative integration of various simulations into the protoKIT instrumentarium has shown a variety of integration possibilities. As expected, multi-agent and distributed simulations have shown to be most suitable due to the compatibility of underlying ontology. In those cases the integration between protoSWARM and specific software has been the technical bottleneck, but no ontological boundaries existed. On the other hand, statistical simulations or simulations involving e.g. static calculation models shown a different class of problems, since the calculations could be made for specific instances of tested systems, rather than be performed continuously.

In all cases, presented work only indicates the general direction for combination of simulations with design systems. Further work is required to research and evaluated specific solutions in this area.

3.5. protoSPACE 3.0

a) Outset

The decision to establish protoSPACE 3.0 laboratory has been made after TU Delft's faculty of architecture has been relocated to its new premises after the fire of 2008. The original concept has been to copy the setup of protoSPACE 2.0 into the new location. However, it has been ultimately decided within the Hyperbody group to re-evaluate the protoSPACE 2.0 laboratory, and propose a new formula for a laboratory for collaborative architectural design[2].

b) Development

The relocated protoSPACE laboratory officially opened its doors in March 2010. protoSPACE 3.0 has followed the relocation of the Faculty of Architecture to Julianalaan 134 in Delft and the laboratory has been installed in one of the former lecture rooms as protoSPACE 3.0. The laboratory had been initially coordinated by dr. MarkDavid Hosale and its coordination has been taken over by the author in July 2011. Initially, the goal set for the renewed for protoSPACE 3.0 was set to "facilitate the continuum between non-standard, virtual, and interactive architecture via collaborative research design systems, the development of embodied interactive architectural components, file-to-factory design work flows, and non-standard geometries in architectural form." This goal has been later compressed to one statement "connect people and things", meaning that the real purpose of protoSPACE_3.0 is to provide conditions where people and things can connect to each other in order to form rich, complex ecosystems, flourishing virtual and physical architectural habitats.

In earlier research, design instruments were seen as developed for protoSPACE 2.0 and were aimed to primarily operate within its boundary. The turn of events and lessons learned from the research changed that perspective. protoSPACE_3.0 can be seen as yet another node in the

[1] Christian Derix et al., 'Simulation Heuristics for Urban Design', in *Digital Urban Modeling and Simulation*, ed. Stefan Müller Arisona et al., vol. 242, Communications in Computer and Information Science (Springer Berlin Heidelberg, 2012), 159–180.

[2] MarkDavid Hosale, 'Protospace 3.0', in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012), 557–560.

Img. 21. Left: protoSPACE 3.0

network of instruments, providing a highly specialised addition to the protoKIT repertoire. Its main purpose is to provide a specialised context in which protoKIT instruments can operate, in which they can be continuously improved and in which new protoKIT instruments can be constructed.

In this way, protoSPACE_3.0 in itself should be seen as specialised infrastructure, provided with a highly-specific agenda. It includes five server computers, a 24-channel surround audio system, five large projection screens, a high-speed local wireless network and a solution for flexibly routing all hardware connections. These facilities provide a “backbone” for operation of virtual and physical architectural multi-agent systems. It can be further extended and connected to other protoKIT instruments operating intensely within, but also beyond the physical boundary of protoSPACE_3.0.

c) Evaluation

The ontological position of a collaborative design laboratory protoSPACE within protoKIT has changed. Originally protoKIT has developed as a set of instruments for protoSPACE. Ultimately protoSPACE has come to being seen as one of the instruments within protoKIT. In this view the network of specialised laboratories being an inherent part of the network of instruments. At the same time, it has been acknowledged, that just like instruments have shown to evolve and adapt over time, being reciprocally influenced by designers and projects they are applied to, a design laboratory does the same. It evolves alongside the projects executed in its premises. In this way, the laboratory consequently becomes two things in one. It is an instrument for development of complex adaptive interactive architecture, but at the same time it also is a prototype of such architecture.

3.6. Network of laboratories

a) Outset

The shift in perception of protoSPACE from a “design process container” to a design instrument allowed a more natural consideration for working with “network of laboratories” in the design process. Different steps in the design process require different facilities. protoSPACE provides facilities focused on early design stages and cooperation between many team members. However, it is limited in respect to prototyping facilities. Consequently two satellite laboratories to protoSPACE have been developed, in the naming convention of protoKIT dubbed “protoFAB” and “protoBUILD”.

b) protoFAB

Since May 2010, CNCdivision had been coordinated by eng. Marco Verde and in July 2011 the coordination has been taken over by ir. Christian Friedrich. The CNCdivision lab of Hyperbody has been developed in close relationship to the protoSPACE_3.0. It can also be considered as another, specialised instrument in the protoKIT. The purpose of the CNCdivision was to expand the possibility of rapid prototyping of physical components in architectural systems.

The CNC division has been equipped with a 60W lasercutting CNC machine with a bed of 90 by 60 cm and with a CNC milling machine with a bed of 300 by 150 cm and two computer terminals for controlling these machines. This has allowed to allow rapid fabrication of virtually developed building components. In recent Hyperbody courses, such as the 2010 edition of the Interactive Environments Minor taught by ir. Chris Kievid, the design process and rapid prototyping using accurate CNC fabrication allowed producing the physical agents of developed architectural systems at the same time as they were being virtually formed.

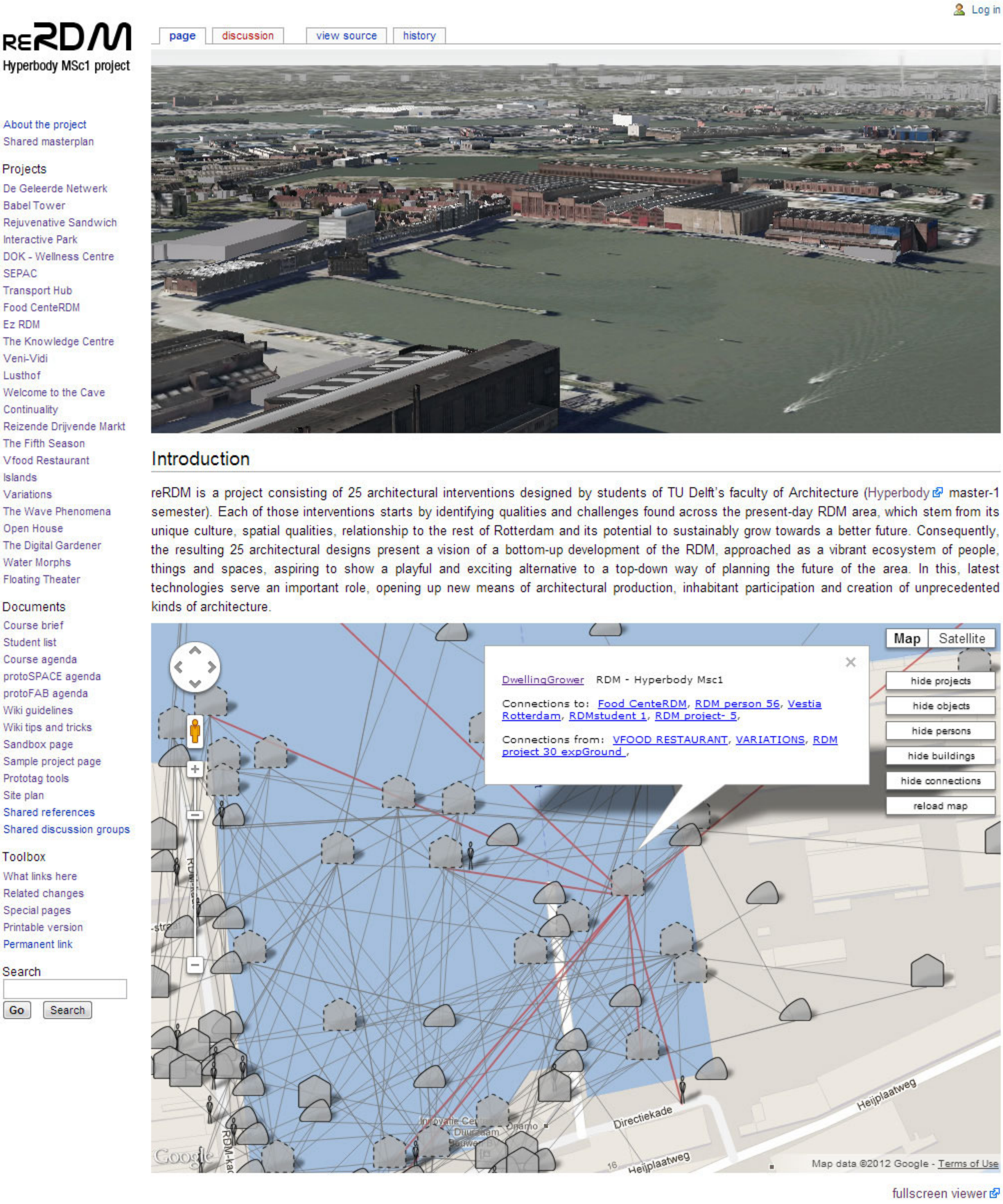
c) protoBUILD

protoBUILD is a proposed name for the officially nameless space provided by the Delft Science Centre to Hyperbody and ID-Studio Lab. It was initiated in 2010, at the outset of the first round of the Interactive Environments minor projects. The projects were built in the main hall of the Delft Science Centre, which was awaiting renovation and being turned into a representative exhibition hall. As the Interactive Environments minor projects have been well received by the visiting public, Delft Science Centre director Michael van der Meer has proposed a more permanent location within the building to host future editions of the course. The shared intention of the Hyperbody, ID-StudioLab and Delft Science Centre staff has been to provide a space where innovative architectural prototypes can be built and tested, while also being a "living" exhibition for the Delft Science Centre visitors. The lab has been designed and built by design firm Tinker, based on input from Michael van der Meer and the author. The approximately 140m2 space had its walls and floor covered with plywood inlay, protecting the walls and floor from damage and allowing for driving nails and screws if needed. A scaffolding structure has been built along laboratory walls and ceiling, providing a generic support structure for future prototypes and storage space for tools and other equipment. A touchscreen has been located on the outside of the laboratory to provide up-to-date information for visitors on projects being built. Visitors can enter and explore projects freely, unless no access is granted due to construction progress. In that case visitors can observe project construction through the glazed wall. At the same time designers can use visitors as test subjects, providing direct feedback on the prototypes.

To date two series of prototypes have been built in the protoBUILD lab. In all cases projects have significantly benefited from the continuous flow of visitors to the laboratory. The projects have been also extensively using the protoFAB and to a much lesser extent protoSPACE. In that respect the distance of approximately 400m between labs has proven to be a hindrance, since fabricated elements and materials had to be manually carried by designers back and forth for assembly and fabrication. Having a permanent location for prototyping, which can simultaneously serve exhibition zone and user testing and research facility has proved beneficial for projects and designers. It allowed frequent iterations of the built prototypes as well as working from "within" of the designed system, blurring the division between the design sketch, mock-up, prototype and the actual structure.

d) Evaluation

Over a course of time, what has been initially envisioned as a single design facility, has become a network of complementary laboratories. Tracing projects from the Interactive Environments minor and hyperbody MSc1 shows that for many projects such network of laboratories easily expands. Students have been frequently using other laboratories of TU Delft, such as Building Technology laboratory for fabricating metal elements, or ID-StudioMake for prototyping electronic components, or InHollabd CompositenLab for fabricating glass- and carbon fibre elements. What's more, several students have contacted commercial firms such as Buitink BV and were allowed to use the company's laboratories for highly specific purposes, such as sealing air-tight materials. Also the growth of protoFAB proceeds in a distributed manner. Two robotic arms, subject to PhD research of Jelle Feringa, have been purchased as an extension to protoFAB, and located in the RDM hangar in Rotterdam due to lack of affordable appropriate location in Delft.



About the project
Shared masterplan

Projects
De Geleerde Netwerk
Babel Tower
Rejuvenative Sandwich
Interactive Park
DOK - Wellness Centre
SEPAC
Transport Hub
Food CenteRDM
Ez RDM
The Knowledge Centre
Veni-Vidi
Lusthof
Welcome to the Cave
Continuality
Reizende Drijvende Markt
The Fifth Season
Vfood Restaurant
Islands
Variations
The Wave Phenomena
Open House
The Digital Gardener
Water Morphs
Floating Theater

Documents
Course brief
Student list
Course agenda
protoSPACE agenda
protoFAB agenda
Wiki guidelines
Wiki tips and tricks
Sandbox page
Sample project page
Prototag tools
Site plan
Shared references
Shared discussion groups

Toolbox
What links here
Related changes
Special pages
Printable version
Permanent link

Search
Go Search

Introduction

reRDM is a project consisting of 25 architectural interventions designed by students of TU Delft's faculty of Architecture (Hyperbody master-1 semester). Each of those interventions starts by identifying qualities and challenges found across the present-day RDM area, which stem from its unique culture, spatial qualities, relationship to the rest of Rotterdam and its potential to sustainably grow towards a better future. Consequently, the resulting 25 architectural designs present a vision of a bottom-up development of the RDM, approached as a vibrant ecosystem of people, things and spaces, aspiring to show a playful and exciting alternative to a top-down way of planning the future of the area. In this, latest technologies serve an important role, opening up new means of architectural production, inhabitant participation and creation of unprecedented kinds of architecture.

DwellingGrower RDM - Hyperbody Msc1

Connections to: Food CenteRDM, RDM person 56, Vestia Rotterdam, RDMstudent 1, RDM project- 5,

Connections from: VFOOD RESTAURANT, VARIATIONS, RDM project 30 expGround ,

fullscreen viewer

3.7. protoTAG

a) Outset

protoTAG originates from a number of problems encountered in projects of the Hyperbody group. Once physical components of designed or prototyped projects are created, their relationships to other system components, including physical components and virtual ones are difficult to trace back once broken. Finding a loose component without explicitly knowing its origin makes it impossible to retrieve any digital information on this object. Similarly, previously existing objects included in the design and enhanced with virtual information don't contain any link to that information or to other components they are virtually connected to in the process. protoTAG has been an initiative of Hyperbody researcher Christian Friedrich to answer to this problem.

b) Development

protoTAG is a system conceptualised by Christian Friedrich. It is a platform for creating direct connections between physical objects and virtual information. The first version has been implemented by Christian Friedrich and Veronika Laszlo, with involvement of the author. protoTAG works by assigning unique, IFC compliant ID numbers to objects, physically tagging these objects and creating corresponding entries in an online database. The tags are linked to those entries using a unique QR-code placed on each label. Upon scanning the label, any authorised individual, can add, change and remove parameters from the protoTAG.

c) Evaluation

ProtoTAG is a system that requires close coupling to the project database. In the version developed to-date the protoTAG database has been protoTAG specific online MySQL database, and has only been integrated with the following protoMAP and protoWIKI instruments. Further development of protoTAG is expected to involve integration with other tools through integrating the MySQL solution with the broader protoBASE system.

protoTAG's use has shown a wide range of possibilities going beyond its originally intended applications. In the reNDSM project it has been used to collect feedback from actual inhabitants of the design site. It has also been used to generate id numbers for virtual entities before their physical realisation. In this it has shown a versatile nature of the platform. However, for the said applications the protoTAG as a printed label in its current form has not been adequate or even redundant. Therefore it can be expected that in the future protoTAG will branch into a wider range of instruments, or some of its current features will be integrated into other applications.

3.8. protoMAP

a) Outset

protoMAP is an instrument which has directly evolved from the protoTAG platform. Having a large number of tags in the reNDSM project required a way to navigate through tags. The original search-based interface has become inefficient, creating external lists of IDs of protoTAGged objects has been vulnerable to error and neglect from the side of designers.

b) Development

protoMAP is a simple solution to the above problem. It is a google maps based online application developed by the Veronika Laszlo and the author as a simple way to navigate through protoTAGs, based on their "last seen at" property containing most recently recorded

Img. 22. Left, from top: protoDECK 2.0 featuring Dieter Vandoren, protoNODE in protoDECK 2.0.

support various current and unanticipated future technical installations and upgrades to the Interactive Experimentation Lab through a modular system of interlocking, fully customizable wooden tiles. protoDECK has thus emerged to be a multi-purpose installation. Initially only intended to hide the laboratory cabling, it has also become a new ubiquitous interface to be connected various future instruments used in protoSPACE. Yet, what's more, it has also become a prototype of a dynamic architectural space, in which various spatial behaviours can be investigated.

After the CNC milling and laser cutting of floor components, performed by the company Nedcam the material part of the protoDECK has been installed. In parallel, dr.MarkDavid Hosale has prototyped and ordered the first version of floor microcontrollers dubbed protoNODE. The 168 active tiles of protoDECK were intended to be controlled by a central PC control unit, collecting sensor data from all tiles and sending lighting commands. However, due to a number of errors resulting from the electrical interference, distance and number of elements, the system has failed to operate.

After the project has stalled the author has taken over the coordination of protoSPACE laboratory and in collaboration with dr. Stefan Dulman and Andrei Pruteanu an alternative behaviour design concept has been proposed. Based on conclusions from earlier discussed projects a fully distributed behaviour of the floor was envisioned. The tiles would no longer be controlled by a central computer, but only locally communicate with each other. A number of possible distributed behaviours has been proposed, including lighting patterns interacting with the public during social gatherings, or dynamically organising the space where for lectures or design sessions. In all cases light patterns were not top-down predefined, but would be generated from local interactions between individual users and tiles.

Local connections were added between tiles on a segment of protoDECK consisting of 20 tiles. Initial applications were tested with successes, including e.g. propagation of a wave pattern of light across tiles. However, it has been concluded that the protoNODE hardware has not been appropriate for intended applications. A new set of hardware has been provided by NXP company and is awaiting its deployment permitting further development of protoDECK.

c) Evaluation

protoDECK project has shown a multi-faceted nature of a design instrument. In this case an instrument aimed to be an interface to the design environment can also become a dynamic actor in that process, explicitly and dynamically influencing the behaviour of designers during the design session. At the same time it is also a prototype of an interactive space, which can be used to deploy and validate the actual system to be further implemented in a different designed building.

Finally protoDECK has also tangibly demonstrated the technical challenges involved in the actual realisation of interactive architectural spaces, which will be further analysed and discussed in the next chapter.

Img. 23. Left: reRDM.hyperbody.nl website integrating protoWIKI, protoMAP and protoTAG connection.

geographical coordinates. Consequently tag “name”, “description” and “front image” properties would be displayed for each tagged with them object in front of a navigable google map. As a result, designers and non-experts alike could easily find tags of the reNDSM projects by navigating through the site map.

As the project was advancing, based on direct feedback from designers more extensions were introduced to improve the usability of the instrument. These included adding templates for newly created tags for objects, buildings, projects and people, where each tag created through such template would contain by default the generic properties for that type, including properties required to view the tag in protoMAP.

Later, a new unified “connection” property type was introduced. This allowed to uniformly link tagged objects among each other and to visualise these connections as lines in the protoMAP viewer.

c) Evaluation

The use of protoTAG has been compulsory to students and a high number of tags has been created in the design process. However, ultimately tags were not structurally used in the projects, with only several exceptions. In the given feedback students explained that at the end of the project they could see the potential of the instrument, but they have not actively used it due to low direct benefit. On the other hand, with few exceptions, inhabitants have not actively added information to the tagged objects for similar reasons. Questions were raised such as “how to make sure that relevant information is recorded on tags?” or “how to design the physical tag in such a way that accidental passers-by understand that interesting for them information can be obtained by scanning the tag and that they can express their own opinion using it?” Further development of the instrument is aimed to address such questions.

3.9. protoWIKI

a) Outset

Various versions of protoBASE, including the database in protoTAG have been focused on recording organised and highly specific project data. Such data is incomprehensible for persons not directly involved in the projects or in many cases also not for designers themselves. At the same time, due to required feedback from users and collaborating designers, current state of every virtually developed project has to be vividly communicated among people. What’s more, it has been observed that preparation of end-presentations, posters project descriptions and other conclusive material of the projects takes significant amount of time of designers, is done at the end of each projects and often lack time to be finalised properly. For those reasons, protoWIKI has been developed and provided as an instrument for project documentation complementary to protoBASE.

b) Development

protoWIKI uses the mediawiki platform. It has been initially used in the VHpark project and was intended as one repository for many future student design projects. There each “atom” of students was given one namespace with a main page and unlimited subpages. Students were asked to update the wiki on a weekly basis and were not allowed to use any other medium to communicate their projects. All project consults and presentations were using the material placed on the wiki. Used in this way, protoWIKI has proven to be a good medium for in-process project communication. However, student feedback and teacher observations have indicated that little communication between atoms has taken place, and as a result atoms have not merged their projects as originally intended.

In the following reNDSM project, the brief has been adjusted to promote collaboration within projects, while retaining their individuality. protoTAG and protoMAP have been structurally integrated in the protoWIKI. The layout of the wiki has also been changed and individual projects were constrained to use the same layout template, both to improve the legibility. Consequently as observed by external evaluators, the quality of documentation has significantly improved. Nevertheless, connections between projects could still have been improved.

c) Evaluation

protoWIKI has proven its role in the collaborative design process as a medium to share and communicate human-legible information about developed projects during project development. The regular updates of the protoWIKI by students were enforced on them, however through practice they became their custom and no complaints were received. The wiki documentation allowed individuals external to the project to monitor the development of projects. Nevertheless, little to none feedback has been collected through protoWIKI discussion pages.

protoWIKI continues its evolution. Based on recent evaluation, next version will include a commenting extension in place of unfriendly for users "discussion" tab, which will be removed. Initially protoWIKI has been planned as a single wiki platform. However, projects documented using protoWIKI are easy to group into clearly defined categories (as VHpark, where projects involved the same location and had the same timeframe) or one meta-project (like reNDSM, where all projects together formed one masterplan). Because of this, it is convenient to split protoWIKI into category specific sub-wikis, which can be bound through hyperlinks and other pages.

A separate aspect of the protoWIKI is the exchange and sharing of technical knowledge, which differently than design content, needs to be gradually built up over longer time. For this purpose a special "tech" wiki has been started up, where entries are grouped per technique.

3.10. protoDECK

a) Outset

protoDECK has been initiated by Chris Kievid and further designed and developed eng. Marco Verde and dr. MarkDavid Hosale, to be eventually taken over by the author. Its original intention has been to add flexibility for the protoSPACE 3.0 laboratory installations by providing an elevated floor, under which cables and other installations could be easily fitted and removed; a feature lacking in protoSPACE 2.0.

b) Development

The project's ambition grew, and in discussions between Chris Kievid and Marco Verde the floor has been conceptualised as a system of individual tiles with a unique design. Additionally each tile was envisioned as device, able to sense steps of laboratory users and change colour and intensity of an embedded LED light. Consequently the geometry of the tiles has been parametrically designed by eng. Marco Verde, and the electronics, sensors and lights were designed by dr. MarkDavid Hosale who joined the team. Eventually protoDECK has been described as "a catalyst as much as (…) an expression of architectural and interaction design. Designed as an open system, protoDECK is both physically and behaviourally a modular system developed to embody multi-modal interaction, and to be adaptable to the research and education needs of protoSPACE 3.0." protoDECK integrates the capabilities of a conventional technical floor, providing a fast solution for the installation the infrastructure needed to

3.11. Discussion

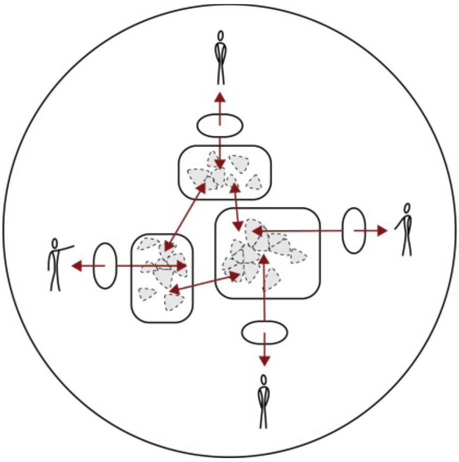


Fig.57. Ideogram: Distributed virtual prototypes constituting a project.

The development from protoSPACE 1.0 software to protoKIT of design instruments has shown a complex evolution and diversification of design “instrumentarium” for complex adaptive interactive architecture, in conjunction with expansion of non-standard architectural design possibilities. The centralised approach towards development of protoSPACE 1.x, despite many promises, has become caught in several bottlenecks where significant time and skill investments were required to further develop the prototyped software. The distributed approach of protoKIT removed those bottlenecks and has initiated an explosion of diverse instruments flexibly combinable with one another, depending on the nature of the given project.

On the other hand, protoKIT, seen as a more general approach, has presented its own problems. The diversification of instruments results in lack of overview and multiple incompatibilities between instruments. Shall this development continue without additional integration, the instruments will become independent of each other, resulting in the scenario known from the commercial sector where interoperability of software from various vendors is minimal and highly constraining.

In order to maintain integrity among protoKIT instruments a binding framework is required that will provide a balance between allowing further differentiation and evolution of instruments and maintaining integrity, coherence and ease of use of the instrumentarium seen as a whole.

Additionally, the link between design instruments and design prototyping has been identified as growing in its importance, as material prototyping begins early on in during the design of iA, while design process as such practically never finishes if the created architecture is to remain interactive and adaptive throughout its entire lifetime.

Conclusions summary:

- An “ecosystem” of design instruments and distributed project database open new design possibilities and increase design process flexibility.
- An extensible framework for design instruments is needed to provide a shared information exchange structure and ontology, permitting further growth.
- Virtual design environment evolves from a centralised virtual space to a network of semi-virtual components.

4. Extending multifaceted development and evolution.

Original idea of protoSPACE was to create a virtual reality environment, in which the design can be developed, simulated and eventually realised. There would be different views on that vr environment for different specialists. The project would develop in that environment. Investigating design case study research has shown that projects also evolve. Such evolution happens across projects, as thy cross-breed mutually influencing each other, or across multiple iterations over one project. However, approaching projects as systems of many components, means that those components can also evolve within projects as they develop, in turn making project's evolution possible even throughout its single development cycle.

This chapter has gone one step further by showing that not only projects develop and evolve, but design instruments do so as well. Over the course of years, the protoSPACE environment being an instrument for collaborative design has distributed and differentiated itself, cross breeding with other instruments and ideas, eventually leading to creation of an ecosystem of instruments for design of interactive and non-standard architecture. Examples have shown that this process should be further stimulated, as it provides continuous increase of quality and robustness of developed architectural systems.

Yet, a risk has presented itself in respect to the evolution and differentiation of projects and instruments alike. Connections need to be maintained between differentiated members of both of those distinct species. Shall these connections be not maintained, compatibility between projects and instruments can quickly decline and evolution will be impaired. Shared conventions can serve the role of the connecting tissue, and these conventions are to be defined in a framework for interactive architecture

Nevertheless, there is the third aspect to the design of interactive and adaptive architectural systems. As shown in past chapters, prototyping of iA systems becomes extensively integrated with the design of these systems. Consequently, design instruments often need to become prototyping instruments as well. Such has been the case with many listed examples. protoFAB and protoNODE are instruments exclusively targeted at prototyping of interactive systems, while protoTAG or protoMAP are instruments that link the virtual with the material. However, also other instruments such as protoSWARM develop connections to the material world. Clearly, before the framework for interactive architecture can be assembled, closer investigation needs of the material processes of prototyping and realisation of interactive architecture need to be made. The process of virtual development and evolution of architectural components, and of instruments facilitating that development and evolution, needs to be extended into the material world of realised architecture.

VI. Tracing materialisation processes

Summary

The sixth chapter focuses on processes of project materialization. Case studies have shown high importance of deployment of working systems in early design process stages. Investigated iA system deployments included virtual and physical simulations, mock-ups, experiential prototypes, or actual realisations. In all cases early deployment proved advantageous, resulting in accelerated project development, exploration of a wider range of design solutions and delivery of more refined architectural qualities. Chapter VI investigates in more detail the problematic of extending virtual design systems into the physical world through embedded technology and rapid component fabrication.

The chapter starts with providing a generic typology of interactive building components based on earlier traced projects. Various prototyping and realisation techniques and instruments are discussed in relation to this typology (section 1.). Consequently the formation of component networks is investigated, in respect to analysing ways in which components can be interconnected in respect to physical connections as well as creation of communication channels (section 2.). Eventually the processes of gradual development of complex interactions between iA system agents across such networks are investigated (section 3.) In conclusion, the integration of design development, design instruments and prototyping instruments is discussed in correlation with challenges of interactive architecture project organisation (section 4.).

1. Materialising building components

Challenges:

- Prototyping activities require structuring and organisation to increase their efficiency in respect to time, costs and skill required, as well as the quality of attained solutions.
- Reusable and extensible typologies of components can accelerate knowledge reuse, focused development and cross-project advancements.

Previous two chapters have traced and discussed evolution of methods and instruments for complex adaptive interactive architecture. In many of the traced case study research projects, the role of performative and experiential prototyping has been critical for project development. Yet to the point no focused attention has been given to the actual prototyping techniques and instruments required in such process.

The prototyping could be split into two broad categories of activities. The first one is creation of the physical building components, either scaled down or full-sized. The second group involves the prototyping of interactions, including the making of sensor-processing-actuation systems, designing their behaviours and interactions, testing and evaluating them.

This section attempts to provide taxonomy of the complete range of building components that have been prototyped in the studied projects. It does so in order to provide more structure to the broad and relatively unexplored domain of dynamic building components. In studied projects numerous practical solutions have been designed and prototyped, however for these solutions to reach the employment in full-featured buildings, their further development and testing is essential. This is required to increase features such as durability, safety, decrease cost and fabrication time and improve the ease of assembly.

Following Usman Haque, "Constructing right from the start erodes distinction between design, construction, modelling and inhabitation. (...) The building is the model."[1] Following this design philosophy, each design has been considered to be a "flat" system without separation between models (representation) of past, current or future reality, and the actuality. Instead of representing reality - being part of reality, and being formed and developed through interactions among its virtual and actual non-living constituents, architects, clients, other experts and eventually also inhabitants. This step is essential to remove the traditional modelling conventions from the design ontology, allowing the concept of the model to be re-introduced and re-interpreted at the later stage of research[2].

1.1. Form

a) Outset

In respect to structural typology, two main kinds of interactive architectural installations have been encountered. The first type, as in the EvoStructure or Digital Pavilion3, starts with the structural members and nodes and in-fills are secondary. The second type, as in Bubble Lounge, D|E|Form or Protosapce 4.04 pavilion uses integrated structural and spatial

[1] Matthew Fuller and Usman Haque, *Situated Technologies Pamphlets 2: Urban Versioning System 1.0* (New York: The Architectural League of New York, 2008).

[2] Jaskiewicz, 'Dynamic Design Matter[s]: Practical Considerations for Interactive Architecture'.

[3] Chris Kievid and Christian Friedrich, 'Digital Pavilion, Intelligent Architecture', in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012).

[4] Christian Friedrich and Owen Slootweg, 'Protospace 4.0', in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012).

components. The adaptation of structures needs to converge of multiobjective optimisation, driven by the usage of a building, structural optimisation of an adaptive/kinetic building system[1,2,], climate and acoustic performance.

The concept of protoCELL integrates the two types of structures encountered in studied installations and provides means to deal with the aforementioned performance considerations in an integrated manner. The term protoCELL refers to the basic building component of an interactive building. In earlier studied projects a variety of such components has been designed and prototypes. In each project, however, such design started from scratch. Often unwarily ending in proposing similar solutions to already earlier design. At the same time, in this manner component designs have never reached a fully robust and market ready state, as projects would be terminated beforehand. It can be hypothetically assumed that if consequent projects would build up upon earlier protoCELL designs and prototypes and improve or fork from these designs instead of starting anew, the performance, cost, production time and other qualities of specific protoCELL types could be significantly improved. In order to permit the iterative evolution of protoCELLs, an overview of the current state of development has to be provided.

The following taxonomy provides an attempt to systematise the protoCELLs developed to this point. Their geometric properties were used to organise this taxonomy. Consequently, four types were identified, namely; a) node components (all dimensions relatively small), b) linear components (one dominating dimension), c) surface components (two dimensions are substantially larger than the third) and d) massive components (all dimensions are considerably large). Other types of component classification are also possible. Among them, organising components in respect to their function is currently most common and included in standards such as the Industry Foundation Classes (IFC). However, functions of building components can easily overlap, or even be changed over the course of their use. Therefore the following general categories are proposed as only a generic indication of component's function: a) structural (supporting not only itself but also other elements, including users), b) space-dividing (physical, thermal, visual, acoustic or any other kind of spatial division), c) space-augmenting (changing the conditions of space; being a part of a larger installation, network or a stand-alone device, providing specific spatial affordances without altering spatial topology). The two taxonomies are clearly indicative and open for interpretation. Rather than delivering any fixed classification model, their purpose is to provide a starting point for experimentation and evolution of new taxonomies, as well as new component kinds, merging and enhancing identified qualities. Following classification presents thus a range of component agents explored by the author, grouped based on geometric qualities, while discussing specific roles in larger systems individually for each example.

b) Node components

Starting from the smallest of considered scales, a "node" component is an entity of a relatively small dimension, or of dimension and shape that make it not spatially perform as a linear, surface or volumetric component. Typical nodes in found in architectural systems can be a) structural nodes b) appliances or furniture elements that don't belong to larger scale categories but which play a role in the given spatial environment c) nodes that are hubs of installation networks.

[1] Werner Sobek and Patrick Teuffel, 'Adaptive Systems in Architecture and Structural Engineering' (July 30, 2001): 36–45.

[2] Patrick Teuffel, 'From Adaptive to High-performance Structures' (presented at the ISSS 2011: 6th International Symposium on Steel Structures, Seoul, Korea, 2011).

o Structural node components

Structural nodes exist in diverse kinds of truss structures, connecting ends of truss members. In complex static structures, such as ONL [Oosterhuis_Lénàrd] Hessing Cockpit and Sound Barrier, nodes can connect members at non-repetitive sets of angles, meaning that each node needs to have entirely unique geometry. In Hessing Cockpit and Sound Barrier over thousand structural nodes have been integrated in the built structure, making CNC fabrication the only feasible option of production. In static structures, an agency of such nodes, seen from structural performance perspective may be difficult to observe, although being hubs of all forces going through the structures, nodes concentrate and distribute all structural stresses in truss structures.

In case of kinetic truss structures, structural nodes become the hubs of motion. Angles in such nodes often need to change dynamically. In Muscle Tower II, Muscle Facade, Muscle Space and Odyssey projects distinct solutions for such problems have been designed and successfully tested. Muscle Tower II[1] required nodes working on both compression and tension, with no more than 10 degrees of deformation for each strut. Hollow ball joints, with bolt connections tensioned by rubber pads proved to be an ideal solution. Muscle facade had joints working entirely on tension. For this metal frames, with easy to attach karabiners proved very suitable. Muscle Space project required a connection allowing a scissor deformation, maintaining the continuity of opposing struts, while allowing high degree of deformation otherwise and additional connections to tensile force actuators. A solution has been found, by connecting two tubular profiles with a bolt and an intermediate plate for additional points of attachment. In the Odyssey installation, a node was needed that would allow regulated flexibility, giving some structural rigidity, while allowing radical deformations. A node made of recycled car-tire steel reinforced rubber, centrally bolted together, proved ideal for the task. The varying of length of individual node arms and the order of their stacking allowed for far going regulation of the required flexibility.

In case of tested projects, structural nodes were not embedded with microcontrollers. There actuation of kinetic structures came from struts, not nodes. However, when controlling behaviour of such structures. However, in virtually counterparts of these systems, autonomous behaviour to individual nodes has proven beneficial. Nodes are natural hubs with multiple connections, where all structural forces intersect. If multiple of connected struts have kinetic actuating properties, nodes can achieve multiple degrees of freedom in displacement. On the other hand, a node is characterised only by one position parameter. Consequently, structural nodes provide good candidates for being embedded with protoNODEs.

o Trans-structural nodes

The project, which may serve as a vivid illustration of breaking established conventions, is the sCAPE group installation, which was built during the Interactive Environments Minor course organised and co-taught by the author. The main idea behind the sCAPE installation is the LEDwork; a network of nodes with embedded LED lights building up an artificial ecology of entities capable of local interactions between each other and visitors to the installation space. In this project two main kinds of nodes have been developed by students.

The first kind is the structural nodes. They function as connectors between struts of the installation structure. Apart from their obvious structural role, they each contain a separate multi-colour light source on both sides of the structure. These lights are controlled by an embedded protoNODE, which is capable of local exchange of simple messages with all other nodes connected to it physically through attached struts. In each of these struts runs a four-

[1] Owen Slootweg, 'Muscle Tower II', in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012), 404–408.

thread wire, where two threads deliver power to connected nodes and two other allow for bidirectional sending and receiving of data. Through the emitted light, these nodes can affect the visual quality of spaces on both sides of the structure. In this way structural performance of these nodes overlaps with their space-augmenting role, followed by space-dividing, as spaces on their two sides become differently illuminated, and therefore more distinguishably divided.

The second kind of LEDwork nodes consists of detachable elements. Each of them contains four strong magnetic connectors with embedded contact rings. In this way detachable nodes can be manually connected to each other as well as to the permanent nodes, which have two additional connections equipped with same magnetic plug. In this way power can be delivered to detachable nodes to charge their internal batteries and data can be locally sent between interconnected nodes. These nodes can be interacted with through touch, rotation and shaking, responding with colour changes of their arms. Their primary role is space-augmenting, while they can have a minor structural function when more nodes are connected through them to the structure. Although not prototyped, many other kinds of nodes have been designed by sCAPE students. These included nodes with moving arms hosted in a flexible envelope, suspended on retractable cables, vibrating or emitting sound.

o Appliance nodes

sCAPE project builds a bridge between structural nodes and device nodes. A device node can be any kind of appliance; a TV set, lamp, smartphone or even a blender. All such devices are typically seen as standalone objects. However, they do have a strong tie to spaces they are placed in. A TV set, clearly transforms the affordance of space, allowing a group of its users to watch moving picture, while at the same time having a strong impact on the spatial organisation of their activities (as a popular saying goes; television has transformed the family circle into a half-circle). Similarly, any other appliance or a piece of furniture may invoke diverse cases of spatial effects through the way it is used.

c) Linear components

Linear components often function as edges forming links between individual nodes. An edge can be abstracted as a line segment. In structural systems edges gain the function of structural members; struts, bars, cables, beams or other. As the main flow of forces occurs typically in a straight line between nodes and only lesser forces appear in other directions (buckling, wind loads etc.), these elements are geometrically stretched; i.e. one of their dimensions is significantly larger than others.

o Struts, ties, beams and columns

Struts are primary building elements of many buildings, especially where complex form and efficient structural performance are required. In complex trusses, struts are replaced by ties where only tension forces exist. Columns can be seen as a specific kind of a strut, having vertical or close to vertical direction and carrying compression loads. Beams on the other hand are structural elements that carry loads perpendicular to their direction, thus typically carrying floors and roofs.

To revert to familiar examples, in ONL's [Oosterhuis_Lénàrd] Cockpit and Sound Barrier project, thousands of struts have formed the structure of the buildings by spanning distances between nodes. No two members have shared the same length, although the profiles were the same across the cockpit part and across the sound barrier part. However, many truss structures are repetitive and modular. With systems such as Octatube erection and modification of truss systems can be highly accelerated, but its modularity limits the diversity and customizability of reached forms.

The “Modular Customisation” by Sjors de Graaf (part of reNDSM), is an examples of a broader group of projects, where actuated or produced to measure linear structural members are a foundation of adaptable architectural systems. In the Modular Customisation system, users can assemble three dimensional structures themselves and each strut locally measures its internal tension force and warns if the structure is in danger of collapsing.

o Linear actuators

In kinetic truss structures, linear elements are typically the ones to drive the movement of the structure. Two kinds of linear actuators have been used for this in case study projects. Festo fluidic muscles have been employed in cases that require fast actuation dealing with tensile forces only and a small displacement stroke, not greater than 10% of actuator length. Additionally fluidic muscles can be deformed by being bent around soft curvatures. This actuator technique has been employed among others in structures of the original Muscle NSA project, Muscle Tower I & II, Muscle Space and Muscle Facade projects. Odyssey project, on the other hand, has employed electric linear actuators. Those actuators are significantly slower, but provide larger stroke, grater force, are much more silent, and don’t require connection to a strong air compressor, as is the case with Festo fluidic muscles.

Considering them as elements of a multi-agent system, such actuators may be treated in two ways. They can be approached as extensions of one of the adjacent nodes, or they can be defined as autonomous agents operating in-between and in connection to the two nodes they span.

d) Surfaces

Nodes and linear components typically function together forming architectural structures. However, due to their point or line geometries, their space-defining qualities are limited. Surface elements need to be introduced in order to define spatial boundaries. A surface is a shape, with significantly large length and width and small thickness (mathematically, a surface has no thickness). In buildings surfaces may be structurally dependant on other components, or can be standalone objects, delivering structural properties. The architectural quality of a surface lays in its ability to block or permit the flow of matter or energy through, being either light, heat, sound, air, smell, electromagnetic waves or other.

The term “membrane” is commonly used in architecture to describe tensile structures. However, taking analogy from biology, where a membrane is defined as a selective barrier between two phases (e.g. a cell membrane), this generic definition of membrane be extrapolated to the architectural scale. Thus in the following descriptions, membranes will be considered to be any surfaces with selective permeability. It is the selective nature of architectural surfaces (e.g. allowing the flow of light, but not matter), that makes the definition of a membrane applicable and underlines the most valid for this research aspect of the performance of architectural surfaces.

o External membranes

The simplest of surface components are typically ones that are used to fill the open spaces between the structural elements (infill). The purpose of these surfaces varies. If on the external layer of a building, its role is to shelter from the climate, thus block rain, snow and wind (flow of matter, including resulting dynamic structural loads) from the outside. It also has to work as thermal insulation, blocking the flow of thermal energy outwards (in a cold climate) and inwards (in a hot climate). It additionally works as an element that sets the boundary between public and private space, regulating the visibility of the inside from the outside, as well as the flow of sound.

The muscle facade project provides a good example of a dynamic external membrane component. In static buildings, facade membranes can be made of glass, masonry or a wide range of other infill or cladding materials. External kinetic membranes are a greater challenge, since the deformation of the facade typically requires deformation of the membrane, not only involving bending, but multidimensional stretching. For this reason, muscle facade membranes, located between the three dimensional network of nodes and fluidic muscle actuators has been designed out of a transparent silicone rubber cushion of a spindle shape, filled with air and tightly sealed. This solution allowed not only a substantial amount of deformation of such cushions; it also provided extensive light penetration and individual illumination of cushions from the inside with changing intensity and colour light. Although the actual cushions have not been prototyped in the muscle facade project due to budget and time restrictions, in the MSc3 design studio by dr. Nimish Biloria, a small scale test for a similar solution has been successfully performed and tested. PTE inflated façade cushions have also been successfully used in numerous realised building projects (such as Cloud 9's Media-TIC building in Barcelona or National Aquatics Center in Beijing by PTW Architects, CSCEC, CCDI, and Arup), yet without extensive deformation capabilities.

The Muscle Facade example illustrates several ways in which a facade membrane element can work as an active component, filtering the passing through it flow of matter and energy and optionally effectuating other qualities, such as being the source of light in itself. This in itself provides a valid reason for treating such elements as agents in the architectural system. Novel, movable components can also be introduced, e.g. in form of façade crawling robots[1].

o Internal membranes

Internal vertical surfaces have a similar role to external ones, but they have to perform over boundaries of lesser magnitude of condition differences on both their sides than is the case with boundaries between building interior and exterior. For internal membranes, sheltering of users from conditions of the outdoor climate is not necessary; however boundary-setting, acoustic, thermal, humidity and olfactory separations, as well as a privacy regulating role can be of significant importance.

In certain cases internal membranes inserted into buildings may be standalone objects. Such membranes do not require any other support than the ground surface they stand on. Their advantage is a non-permanent attachment to the location in a building, thus also ease of modification. Logically, kinetically dynamic surfaces that belong to this category are least constrained to their environment, therefore easiest to develop generic prototypes of. Described further examples are all standalone installations, mostly due to their temporary, experimental character.

The curtain project's concept was to create a membrane surface that would allow appearance of openings of various sizes in any point on its surface. The idea for achieving it was to compose the surface of a high number of vertical, bendable elements that would locally deform by bending sideways to create dynamic openings. The rods constituting the surface were not active by themselves, but were deformed by a chain of plates mounted on top of them and holding their top in place. A frame structure has been built to support the membrane system in any location, however the surface by itself could have been easily integrated into most building interiors as a more permanent element. Each panel segment's angle of rotation was independently controllable. Each of such segments was also equipped with two ultrasonic proximity sensors pointing to both sides of the structure. In this way, each such segment was operating as an autonomous agent (although behaviours of all segments were programmed

[1] Stacey Kuznetsov, Eric Paulos, and Mark D Gross, 'WallBots : Interactive Wall-Crawling Robots In the Hands of Public Artists and Political Activists', *Human-Computer Interaction* (2010): 208–217.

on one, shared control unit computer). The prototyped setup consisted of 12 segments; however the system is fully scalable, both in terms of its length, as well as a possibility of adding more layers of its depth.

The second project built during the same course is the leafs portal. In this case, the portal is composed of multiple kinetic surfaces (five in the prototyped setup). Measuring 1,5 by 4m surfaces lay flat on ground in their relaxed state. From this state they can fold up to the limit of full 360 enclosure, twist sideways and rotate around their base. Each surface has four proximity ultrasound range sensors embedded in their base and an accelerometer sensor in the tip of the surface to feedback its accurate position at any moment in time. Their deformation is caused by four independent DC motors, which wind up the cables running along composite glass fibre rods forming the skeleton of each surface. This skeleton is further cladded with by 12 rows of scales made of 1,5mm PVC sheet material. An additional motor controls rotation of each element. Control of each element is handled by a laptop computer running max|msp environment in which the behaviour is defined and communication with other elements can be accordingly set through the local wireless network. In this project each surface clearly operates as an autonomous agent. It can complexly deform itself, including spatially disappearing by unfolding flat on the ground. It is however the interplay of multiple such surfaces that can generate complex spatial behaviours and explorations various aspects of membrane as a portal between different spaces, including transforming one space into another. Both curtain and leafs project have been destroyed in effect of the fire of the TU Delft Faculty of Architecture on the 13th of May 2008.

The Interactive Wall project[1] developed by Hyperbody in collaboration with Festo has builds on a similar premise, of creating a dynamic wall structure that can deform based on interactions with users, environment and between wall segments, as well as through wall's internal behavioural process. Wall segments utilise Festo finray principle, allowing for a complex curvilinear deformation using only one actuator. Proximity sensors and LED light embedded in each unit allow for sensing of people's presence and additional articulation of wall's behaviour.

GEN installation built during the Interactive Environments Minor of 2009 consisted of three types of elements, one of them were the dynamic wall pieces technologically inspired by the InteractiveWall project. The concept for this component was to enhance the quality of the wall surface by allowing for a more tactile interaction. Elements were padded with foam. In combination with the finray effect principles, of shape resistance increasing with the amount of surface being pressed against, it allowed for visitors to the installation to use wall segments as seating elements that would gently wrap around the shape of a person while gradually increasing their resistance and ultimately providing a comfortable seating form. The way in which the wall segments were actuated has also been changed, allowing for more dramatic deformations, with the large displacement ranges of each element's tip.

In projects exploring the idea of a dynamic membrane show to this point it has been clear how agency of is being distributed, as each surface has been clearly divided into modules, each with its own set of sensors and actuators. This may appear less obvious in the example of the Muscle Space project.

The Muscle Space project, introduced already in the earlier chapters, consisted of three surface structures objects non-permanently fixed to ground and not suspended otherwise. Each of these surfaces consisted of a scissor grid of flexible tubes, forming a double curved shape. Festo fluidic muscle actuators have been placed horizontally between joints. Their contraction was locally resulting in straightening the surface in horizontally and due to the

[1] MarkDavid Hosale and Chris Kievid, 'An InteractiveWall as a Prototype for an Emotive Architectural Component', in *: IA#3*, ed. Oosterhuis Kas, Owen Slootweg, and Xin Xia (Heijningen: Jap Sam Books, 2010).

scissor effect reducing its local width and increasing its height, while causing it to bend outwards at the same time. Variation of contracted actuators would result in complex and rich deformations. Such system has been smoothly covered with flexible lycra fabric, forming a continuous surface around the flexible skeleton. Over this fabric dynamic visuals were being projected, enhancing installation's performance.

The question remains, whether the entire shown surface system is to be defined as one membrane entity or whether the actual membrane is only its subcomponent - the flexible surface with projection. In the second case, the shown kinetic surface should be studied as a system built up of nodes, flexible rods and muscle actuators. Arguably this classification depends on the scope of interest. Looking at the scale of a building, it may only be relevant to see the surface as a whole as one entity, a dynamic membrane element with its own behaviour and agency. However, while studying particular, subtle behaviours and interactions on the individual user level, the system needs to be broken down into smaller elements, thus individual nodes, rods, actuators and installed on them passive membranes following the deformations of the structure. Therefore the standalone membrane developed in the Muscle Space project can be considered to be an agent; however this agent at the same time nests a sub-system of multiple agents, being individual nodes, rods and passive membrane patches that constitute the entire object.

- Floors

A specific type of a surface is floor. Floors can be either internal or external membranes, however, other than having the properties of such membranes, they additionally have to support significant dynamic loads coming from the use of the space, including the own weight of its users and all movable objects placed on them. These qualities make it significantly more difficult to achieve kinetic deformations in floor systems.

A floor is a surface that not only forms the bottom boundary of a space, but which in many cases divides architectural spaces placed on top of each other. As much as all demonstrated to this point structures could have been realised using light structures, floors require much sturdier solutions. Typically floors in buildings are made of concrete slabs supported by concrete or steel beams, or, especially in historical buildings, of wooden planks spanning the space between wooden structural beams. A floor is commonly expected to be static. Two approaches may be in employed to break this premise. Either the entire floor structure can be made dynamic, or a static layer can be augmented with dynamic features.

A common example of an entire floor segment that can be displaced is a suspended bridge. It also clearly illustrates the required amount of additional structure and energy required for such enterprise. An example of analogical displacement of entire building floors, capable is the David Fischer's Dynamic Tower project, where entire floor slabs can be rotated around the central core. Another solution, with higher degree of possible deformation has been proposed by Sergio Araya, Duks Koschitz, Orkan Telhan, Alexandros Tsamis in the hiDrone project, where individual segments of the floor space could displace vertically against each other, while maintaining mutual structural support. These projects, nevertheless still remain as only conceptual proposals, not empirically proven, mostly due to the high cost and technological complexity of proposed solutions.

Examples empirically evaluated within this research belong to the second kind of floor surfaces, where floor kinetic floor elements form the top layer on a static base. In this way the behaviour of the top side of the floor does not have any direct effect on the bottom side of it – the ceiling of the space underneath (if any).

The Muscle Re-configured has been a student project coordinated by dr. Nimish Biloria, where a set of dynamically folding surfaces was proposed as a way to create a dynamic building space. The most innovative of proposed elements was a floor segment, which placed on a hard underlying surface allowed for substantial deformation of the top panel made of aluminium polymer composite hi-lite, enforced on its bottom by high density EPS wedge-shaped blocks, stiffening the element while deformed. Multiple shortcomings of this solutions have been identified though, namely relatively high instability of all forms in-between the fully folded and unfolded states, only concave folds possible and only one-directional surface deformation.

A seminal project of a different approach to creation of dynamic floors is ADA from ETH Zurich[1]. In this project each of the hexagonal floor tiles has been equipped with multicolour dynamic light source and a pressure sensor, as well as an autonomous, self-learning behaviour, tuned to develop playful games with installation visitors. In this approach each of such tiles can be seen as “pixel” in the floor system. Author’s conceptual design of an Emergent Playground project illustrates a possibility of developing a similar setup, where each of the tiles would additionally be kinetically vertically actuated (as well as incorporate a wider range of sensors and effectors, including sound and touch).

A different way of applying a similar idea can be found in the component developed by the GEN group in the Interactive Environments Minor 2009[2]. In their setup the floor has been divided into square elements, on top of which an inflatable element has been placed. In the original setup, all elements have been covered with one flexible surface material and equipped with pressure sensors to determine the usage. These components have been extracted from the project and evolved into Cloud10 installation, where an improved version of cushions has been set up as a dynamic lounge space.

e) Volumetric components

The GEN lounge floor system is another example, where the scale of the scope of view determines the classification of the component. When seen as a whole, the floor of this installation forms a floor surface. When looking at each element individually, especially apparent in the Cloud10 reiteration, each such component becomes a volumetric object. Its length, width and height are of comparable dimensions.

Regardless of their scale, two kinds of volumetric components can be distinguished in common architecture. The first type are rigid, typically monolithic elements. An example can be a thick concrete or stone wall, the dimension of which has significant depth. Ancient pyramid walls may be the most explicit example here. Such components distribute forces and energy in multiple directions. As much as from the structural engineering point of view such heavy structures are seen as far from optimal in small scale buildings, in structures built for extreme conditions they may often be necessary, usually in form of monolithic reinforced concrete elements. Such massive components may also have certain additional qualities, such as high, equally distributed insulation, acoustic absorption and capacity to accumulate energy within their mass, which results in raising the temperature within building during cold days or at night and cooling down during hot weather (this property is taken advantage of in the Trombe Wall, with heat collecting properties enhanced by the placement of a glass panel in front of a massive wall). Such monolithic elements, usually of high mass, are inherently inert, thus the main domain of their application is creation of inert, static parts of buildings and support of other components. Their dynamic performance typically ends with the moment of their materialisation. Changes or demolition of such elements requires substantial expense of energy and time.

[1] Eng et al., ‘Ada - Intelligent Space’.

[2] Jaskiewicz, Van der Helm, and Aprile, ‘Creative Approach to the Design and Prototyping of Experimental Smart Spaces, Case Studies from the Interactive Environments Minor’.

This cliché has been challenged by the Protospace 4.0 pavilion[1] designed in collaboration between Hyperbody students and researchers. This pavilion's design is an irregular form, built up of over 200 massive components, of varying dimensions, around 2.2m by 1.5 by 0.5m. The elements are made of an EPS core, coated with a strong polymer, making it a strong and robust, unique composite material solution. Each of such components, parametrically designed, is unique in its form. Its method of fabrication using a combination of cnc hot wire cutting and CNC milling allows not only for fast production, but also for quick disassembly and almost complete recycling of materials. Although kinetic deformation of components themselves is not possible, each of them has an embedded controller and communicates power and data through it to all its neighbours. The design allows for extension of component with a wide range of devices and appliances, including HVAC units, lights, displays, furniture pieces (with optionally kinetic elements), as well as insertion of openings for regulated daylight and visibility connection with the outside world.

The second type of volumetric components is non-rigid ones. They typically consist of the envelope, mechanical and/or electronic parts responsible for their functionality and flexible filling-in substance. Such is the case with GEN floor units, which are filled with a regulated amount of air, held by recycled cargo bags and mounted onto wooden pallets along with controlling circuitry, sensors and Festo air valves regulating air intake and exhaust. At greater scale, Muscle NSA installation is an illustration of such a volumetric dynamic object of a much greater scale. An inflated sleeve is wrapped by a grid of muscles with sensors attached to connecting nodes and a control system placed inside. The shape dynamically deforms in response to activities of exhibition visitors around it, while clearly having a life of its own; an autonomous agency. Similarly to earlier discussed cases, this project can serve as an illustration of a possibly nested nature of such systems. The same muscle NSA installation can be also considered as a system of autonomous nodes augmented with sensors, linear actuators connecting these nodes and a surface of the air tight membrane filled with a volume of air of a higher pressure than in the surrounding space. This air can be thus either defined as a massive, volumetric component, or, surprisingly as building space. It is possible to enter the Muscle NSA's "cocoon" through a concealed air tight hatch, turning its inside into a possibly usable space.

f) Evaluation

In all presented provisional types of physical components; either nodes, linear-, surface- or volumetric elements, the same dilemma applies. In a virtual model, each component can easily be created as an inherently dynamic and fully transformable agent, there is no explicit production cost or time required for any of its virtual transformations to occur, each agent can have an individual behaviour and memory of its own at no explicit expense (except for the limitation of involved processing time and the corresponding capability of employed hardware - a limitation relatively easy to overcome). This allows for creation of rich and complex multi-agent spatial models in virtual environments. Transformation of these models into the physical space faces many more constraints. Agents' ability to change form or create other dynamic effects can be in some cases engineered, each agent can also be fitted with an embedded microcontroller that can locally host its relatively complex behaviours, however, many of virtually easy to achieve transformations are entirely impossible in the physical world, while others come at a high price (either financially, engineering difficulty wise, or time wise). For this reason another useful classification of agents can be proposed, which is related not to their spatial performance, but to the frequency and kind of transformations they undergo. On one end of its spectrum, entirely passive agents can thus be situated, that are static elements, with no explicit behaviour, other than reactions coming from their inherent material and

[1] Chris Kievid, 'Protospace 4.0 Pavilion', *Zeppelin* 92 (2011).

mechanical properties in response to external stimuli. On the other end of the spectrum are advanced embedded agents, which can undergo complex physical transformations, and which can communicate with their environments across various modalities. In the middle of the scale are agents that can undergo some transformation, being inherently active to some degree, but the autonomy of which is limited. An actuator (with or without an embedded microcontroller) steered by an "intelligent node" may be an example of such element.

This classification also leads to the question of hierarchy. Since many of such employed agents are only reactive towards other agents, their behaviour is to a high degree controllable by other agents of higher autonomy. However, they do not necessarily have to become leaf nodes in the system network. An illustration here may be a linear actuator, in this case operating as a simple reactive agent and to nodes it bridges that use powerful embedded microcontrollers and try to obtain certain local position in space. In this hypothetical setup each of them can send a message to the actuator to either increase or decrease its length with a certain specified distance and the actuator will average these requests. Even though this actuator only reacts to sent commands, can thus react to requests from both nodes, unlike a leaf node in a centralised or decentralised network that can only have one controlling node above it.

1.2. Fabrication

a) Outset

The presented overview of dynamic, adaptive and adaptable building components shows a wide range of possibilities. All components have been designed as individual elements; therefore CNC fabrication plays a critical role in their attainability, both in respect to speed and cost of that fabrication. Three aspects can be outlined. Firstly, the fast and inexpensive fabrication is required for the iterative prototyping process. Secondly, many adaptive scenarios require fabrication of new building components throughout the entire building lifespan. Thirdly, accuracy and cost efficiency in respect to component individualisation cannot be achieved with non-CNC production methods[1].

b) Process

The process of introducing CNC-fabrication to case study research experiments has been gradual and dependent on the available facilities, skills of their operation and integration with the design methods and instruments. Prior to the establishment of the CNC division by Marco Verde in 2010 (later renamed to protoFAB), CNC has not been directly used in any prototyping at the Hyperbody group. Lasercutting of small prototypes has been the only exception. The use of CNC fabrication has been extensive with ONL projects (Such as Cockpit and Sound Barrier) at the industrial scale, where the CNC fabrication was performed (by Meijers Staalbouw for steel, by Pilkington for glass elements) only after the design process was finished. In other projects of ONL, such as the TT-monument of F-side, materialisation process has been developed in parallel to the design process. As an example, in the F-side project polyurethane coated CNC-milled EPS component prototype has provided unsatisfactory quality and too high cost and as a result the housing building has be redesigned to use aluminium panels cut with a water-jet cutter and bent to achieve similar form articulation.

[1] Marco Verde et al., 'Investigations in Design and Fabrication at Hyperbody', in *Fabricate: Making Digital Architecture* (London: Riverside Architectural Press, 2011), 98–105.

Eventual location of the CNC division adjacently to protoSPACE has provided a possibiliy for closer integration of design processes with CNC-driven prototyping. The special role in this has been played by the large 3-axis CNC milling machine. Milling elements out of MDF, plywood or EPS blocks has proven to be an affordable and fast method of prototyping. The influence of the direct availability of the CNC laboratory for the designers can be observed by comparing the results of the interactive environments minor 2009 and 2010 projects. Although the teacher input might have played a limited role, both courses have been set up based on same principles and were given similar design tasks. As a result projects from the 2010 semesters delivered a significantly higher quality, and architectural form refinement. Most significantly, access to the milling machine and instruction in design technique for fabrication of complete forms built up of individualized components, has also had an influence on design decisions. Students were inclined to build prototypes taking full advantage of technology at hand.

In Hyperbody Msc1 the scale of projects has been larger and less focus has been put on prototyping the designs. However in 2011 development and evaluation of a building component prototype has been introduced as part of the main assignment, yet without providing detailed instruction on the CNC-milling techniques and modelling for fabrication technique. Consequently students have chosen much less efficient ways of producing. Most chose to use hand tools. Consequently prototyping took extensive amount of time; prototypes were not well corresponding to the virtually designed buildings and, most importantly, the component fabrications were not scalable to industrial production. The opposite was the case in Interactive Environments projects, where the CNC milling process and MDF could be with relatively small adjustments replaced by different materials and machines, such as steel or aluminium and water-jet or plasma cutting CNC machines.

The latest extension of the protoFAB is an undertaking coordinated by Jelle Feringa, involving direct employment of CNC robotic arms witch changeable tools in the integrated design and prototyping process. The employment of robotic arms allows investigating more innovative production methods in pair with computational techniques for generating architectural geometries in conjunction with fabrication constraints.

c) Evaluation

The role of CNC fabrication proved critical in design processes in which building prototypes has been an integral part. It has provided an increase of speed, accuracy, quality, and most importantly it is scalable for further project development beyond the early prototyping phase, since employed CNC techniques are similar to those used in the industry for realization of mass-customizable elements.

The introduction of CNC fabrication and gradual advancement of used techniques has shown to have a direct influence on the designed architectural forms. The availability of facilities and techniques and “way of thinking” developed by designers in the process of using them has not only encouraged the integration of cnc processes into design aesthetics. It has also induced use of “fluid” geometries and has inclined designers to include scenarios of project adaptation by reconfiguration over the time of its use. However, it has not had any direct influence on involved dynamic interactions.

1.3. Controllers

a) Outset

Examples are known (e.g. Theo Jansen's "Strand Beesten"[1]) where purely mechanical systems allow creation of an interactive dynamic devices and installations. However, mechanical approach is highly constrained and requires extensive amount of space and material. Therefore, the use of digital technology is an obvious alternative to mechanical solutions. In any digital dynamic system a controller is needed to execute and coordinate the behaviour of the entity, by gathering data from sensors, triggering effectors and communicating with the outside world, leading to systems-embeded intelligence in architecture[2].

b) Process

In the Muscle projects a PC computer has been used as the controller. Such solution has many advantages, as a PC is a versatile system. Virtools[3] has been used to program the behaviour of the controller in an easy to comprehend and adapt manner. PC-controlled extensions such as a phidgets[4] board, a relay unit or a midi interface were used to connect the PC to a variety of sensors and actuators. This approach has shown to be versatile, as various solutions could be easily tested and using a pc as a platform makes it easy to adopt a variety of techniques and find reference material. Using sound, video projections or graphical interfaces has been also relatively easy and frequently employed. On the other hand, the PC-controller provides a significant constraint in three ways. 1.) Created installations are not scalable. Every interface, sensor and effector needs to be directly connected to the PC, and the number of connections is limited by the PC computational power. 2.) The PC is a large device with high power consumption and long re-booting power. Even though most installations built using a PC controller attempted to work directly upon powering on the PC, eventually in all cases an instructed person had to be present to turn on the system and respond to unexpected errors resulting from complex software setups. 3.) Reconfigurability and extendibility of installations are complicated to realise wit PC controllers. Therefore only systems with predefined set of elements were developed.

In consequence of the limitations of PC-controlled systems, the concept of protoNODE has been introduced. "protoNODE" a microcontroller embedded in the component of the prototyped or realised installation or building, operating as an "embedded agent". "An embedded agent can sense the environment through its sensors and act upon the environment using its actuators. The agent decides how to relate its sensor data and internal state to actuator commands in such a way that its goals are satisfied. Broadly speaking, research in embedded agents concentrates on the realisation of artificial agents strongly coupled with the physical world. Because environments and users of systems continuously change, agents have to be adaptive."[5] protoNODE is intended to be easy to use and program, while allowing a distributed alternative to centralised PC-controlled systems. For initial experiments, such as in the Bubble Lounge project initially Wiring boards were used by the author. Eventually similar to them Arduino[6] has been chosen as both hardware and software platform for protoNODE.

[1] Janssen, 'Strandbeest'.

[2] Henriette Bier, 'System-Embedded Intelligence in Architecture', in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012), 292–303.

[3] '3DVIA Virtools - Dassault Systèmes', accessed July 29, 2012, http://www.3ds.com/products/3dvia/3dvia-virtools/.

[4] 'Phidgets Inc.', accessed July 29, 2012, http://www.phidgets.com/.

[5] Hani Hagras, Victor Callaghan, and Martin Colley, 'Intelligent Embedded Agents', *Information Sciences* 171, no. 4 (May 13, 2005): 289–292.

[6] 'Arduino', accessed July 29, 2012, http://www.arduino.cc/.

In interactive installations of Hyperbody, since the interactive portals projects, commercially available Arduino hardware was used. A special custom version of protoNODE, dubbed protoNODE_1.0 has been designed and produced by dr. MarkDavid Hosale for the use in the protoDECK project. protoNODE_1.0 has used ATMEGA8 and Atmega168 controller chips, which corresponded to the simplest version of Arduino hardware. The author has used an Atmega1280-based "Seeduino Mega" boards as protoNODES for the D|e|Form installation. A variety of other Arduino-compatible boards have been used in student projects, such as Arduino-mini in the Ledworks.

Arduino microcontrollers provide a number of input and output pins (depending on used microcontroller) for physically connecting sensors and actuators. The functionality and number of pins can be extended with additional "shields" boards that can be individually attached or even stacked in larger numbers on each microcontroller. Each Arduino contains at least one hardware UART serial port, which allows local communication between two nodes or between a node and a computer, while e.g. Arduino Mega boards contain four of such ports.

The use of Arduino boards has been common in Interactive Environments minor and frequent in the MSc1 projects. The biggest advantage of choosing this platform has been its widespread user community. Consequently it has been easy for designers to seek assistance and reusable solutions on online forums and user groups. At the same time, abundant projects and technical solutions presented online have also provided inspiration to students.

c) Evaluation

Arduino-based protoNODES have allowed creation of a new kind of prototypes. The small size of controller boards allowed for their direct integration in prototyped components. The user community has accelerated and stimulated the incorporation of a wide range of software and hardware solutions and provided inspiration for innovative technical possibilities.

On the other hand, certain aspects of prototyping have been substantially more difficult or entirely impossible when using embedded microcontrollers alone. Among them are screen based interfaces, quality audio, video display and projection, and web interfaces. For such applications PC computers remain necessary. The d|e|form project has consequently shown the potential of integrating a networked approach where a PC-controller was used alongside a network of protoNODES. Further miniaturisation of a PC controller is nevertheless a desired quality. With upcoming micro-PC systems such as Raspberry-Pi[1] microcontrollers with PC qualities have the potential to be added to the protoNODE family.

1.4. Effectors

a) Outset

An effector is commonly defined as a device "used to produce a desired change in an object in response to input"[2]. Effectors include actuators, devices that produce kinetic transformation, but also include light, sound and other forms of output. Effectors directly influence the kinds of dynamic transformations attainable by building components. Their features also constrain the design of building components. Therefore the choice and availability of effectors for prototyping has significantly influenced the development of prototypes and designs. In studied projects effectors were used both with PC-controllers and embedded protoNODEs.

[1] Eben Upton and Gareth Halfacree, *Raspberry Pi User Guide* (John Wiley & Sons, 2012).

[2] 'Effector', *The Free Dictionary*, n.d., http://www.thefreedictionary.com/effector.

b) Process

- Linear and rotary motion

The earliest interactive projects of Hyperbody have focused solely on the use of actuators, specifically the Festo Fluidic Muscles[1]. The Fluidic Muscle actuators work using compressed air. Once inflated they produce a pulling force of 6000N, and displacement of 25% of customizable length of the actuator. Fluidic muscles have provided fast actuation and allowed an extensive flexibility of use. Several projects such as the Muscle Trans-Ports also took advantage of the actuator flexibility, where bending of the actuators around the inflated core was essential and would not have been possible with any other existing actuator type. Other projects, such as Muscle Tower II, took advantage of Fluidic Muscle speed of operation. Such combination of speed and force could have only been paralleled by hydraulic linear actuators. However in that case it would have required a hydraulic system significantly more expensive and difficult to install and maintain than compressed air system.

Each fluidic muscle requires two valves, allowing inflation, release and holding the tension of the actuator. In studied applications the actuators could only be fully extended or contracted, as the regulation of in-between states has been technologically too complex and expensive to implement. In muscle projects all valves were contained in one control box with the controller PC attached to it and air tubes running to all individual actuators. The Bubble Lounge prototype included an alternative setup, where valves and actuators were locally placed in each component and directly controlled by local protoNODEs and air pressure was produced centrally.

Despite their advantages, fluidic muscles have also presented a number of problems, ultimately limiting their architectural application. 1.) The requirement for compressed air has been a significant constraint. Portable air compressors are not suitable to support larger structures such as Muscle Tower II. Large air compressors significantly increase installation cost and due to weight going beyond hundreds of kilograms they also reduced the portability of installations and added a significant volume of needed space that would not contribute to architectural qualities. The noise of compressors has shown to be a significant hindrance to users and required placement of compressors at larger distances from the used space. The speed of actuation proved an advantage in behaviours such as Muscle Trans-ports, however was less desirable in projects such as Muscle Space, where fast movements of wall elements were reported as “scary” or “unpleasant” by the users.

In consequence, the following iPortals and Interactive Environments projects investigated employment of electric actuators. Two iPortal projects used electric motors to achieve linear actuation through winding up steel cables. The curtain project included eight 12V motors controlled by a PC-controller with a relay unit. The motors were placed in the base of the element and actuation occurred in the top part of the installation transmitted through shift cables. In the “leaves” installation, a similar form of actuation was used, where cables were used to bend the glass-fibre rods, resulting in curling up the element.

In other projects, such as the Odyssey in the Interactive Environments minor 2009, electric linear actuators were used. Unlike the fluidic muscles or cables, electric linear actuators allowed both tension and compression while generating high forces of up to 6000N. Through an H-bridge circuit they could be controlled directly by a protoNODE or a control PC. Their speed however has been significantly slower than of Fluidic Muscles. In case of the Flexscape project (Hyperbody MSc1/2011) a custom electric linear actuator has been engineered by the students to drive the relative displacement of the components.

[1] ‘Fluidic Muscle DMSP/MAS’ (Festo.com, 2012), http://xdki.festo.com/xdki/data/doc_ENGB/PDF/EN/DMSP-MAS_EN.PDF.

Several projects have used servo or stepper motors for small scale actuation. An example is the Odyssey project, were 18 servo motors triggered minute deformations in the foam-covered floor, giving it an impression of a complex, deliberately disorienting wave motion.

- Volumetric actuation

The purpose of linear actuators is to produce force and add kinetic transformation to other parts of the components. However in several traced projects kinetic actuation has been integral to the building components. In all studied cases this has been achieved through employment of pneumatics.

In bubble lounge project the individual components were inflated cushions with Festo fluidic muscles inserted inside the cushions. In that case the pneumatic tension of the component was providing the opposite force to the muscles.

The Spacebook project proposed and prototyped actuators similar to Festo Fluidic Muscles, but much larger in diameter. Consequently, no other elements, but two layers of actuators were needed to construct a solid, kinetic architectural canopy.

The Gen project prototyped a solution eventually continued in two iterations of the Cloud project (Cloud-9 and Cloud-4). The installation floor was filled with inflatable cushions made of cargo bags. Inflation of individual bags allowed various configurations of the floor geometry, permitting various seating arrangements and organisation of walkable zones.

In the Aeolus project of the Interactive Environment Minor (2011), a different volumetric inflatable solution was provided. There wall elements made of light synthetic fabric provided spatial boundary when inflated by integrated fans. When deflated, the elements would be reduced to the fans and fabric loosely lying on the ground. However, the boundary was only visual. Components had no structural capacities, nor were able to insulate acoustically or thermally.

- Light

In pc-controlled installations, such as Muscle Space, usage of computer controlled projection has been the easiest way to implement a flexible lighting setup. In Muscle Space two beamers were used projecting ripple patterns on the installation walls. This approach has had several shortcomings. The main constraint has been the distance between the projector and lit elements, which provided significant spatial constraints in setting up the installations and would have been impossible in many real world scenarios. Additionally shadowing when users or objects were placed between projectors and projected surface has been a significant drawback. Eventually, poor visibility of projected light in daylight situations, high cost, energy use and limited lifespan of projectors has ruled out this solution from prospective applications.

Integration of LED lights in installations has provided significantly more robust effects, as in many Interactive Environments projects. It has however constrained the amount of detail easily attainable. To achieve stronger lighting effects, DMX controlled stage lighting has been used. The Modii project of interactive Environments Minor has used EL wired with success, which proved more difficult to control than LED lights, but provided an alternative lighting quality. Other projects such as Ledworks, protoDECK 2.0 or iLITE[1] show more possibilities of largely distributed light patterns locally interacting with users.

- Sound

The use of sound has not been extensive in studied projects. This might have been partly caused by the lack of skills among architectural designs, and low association of sound as an architectural quality. Among built installations sound generation has been designed as

[1] Chris Kievid et al., ‘iLITE, a Creative Vision on Architectural Lighting Solutions’, in *iA#4 : Quantum Architecture* (Heijningen: Jap Sam Books, 2011).

centralised using a control-PC, e.g. in Muscle Space. However, in these installation sound was treated as an add-on, not an integral part. Author's experiments included simple tests of integration of piezo speakers, giving basic audio feedback in component embedded arduino boards, however they have not been integrated in any of the installations.

Many projects have produced initially undesired sound effects. Muscle projects have been known for the loud noise of air intake and release. In the Odyssey project (Interactive Environments Minor), the servo-actuators integrated in the floor created loud, high frequency scratching sound. The design team has decided to incorporate that sound into the installation and integrate it in the design narrative of an "otherworldly experience".

Not only active sound generation, but also passive sound qualities were given little attention. Among the multitude of projects only few consciously engaged in acoustic design. Among those the "Soundscapes" project of Msc1 Hyperbody 2011 is one of the few exceptions. There the passive and active reflection of externally generated sound has been the topic of design investigation. However the built prototype has failed to present an experimentally verifiable solution.

c) Evaluation

The investigated projects have included inexpensive, off-the-shelve effectors to create complex dynamic transformation of prototyped structures. It has been clearly demonstrated, that a wide range of effects can be created that are typically not associated with architecture. In combination with a wide range of accepted solutions (such as escalators, automated doors or shading devices) this provide a wide repertoire of building actuation and effecting techniques.

However, this repertoire can be further extended by w broad range of "smart materials" not investigated in the scope of this research. Among those shape memory alloys[1] or electro-active polymers[2] or smart glass are only some of many possibilities.

The significant drawback for designers employing effectors in their projects is the lack of access to knowledge required for design and prototyping. Architectural material databases[3] [4] or overview publications[5] typically include few or none active material solutions. New knowledge exchange platforms are expected to emerge in the coming years, providing better instruments for such exchange and permitting community-driven explorations into new areas.

1.5. Sensors

a) Outset

Sensors are required in order to build up complete feedback loops between building components and their environment. A wall-mounted switch might be the simplest and most common kind of a sensor used in architecture. Another kind are passive and active infra-red sensors, often used to trigger alarm, automated doors or escalators. However, a wide range of other sensors exist and can be integrated in interactive building components.

[1] Charlotte Lelieveld and Liek Voorbij, 'Dynamic Material Application for Architectural Purposes', *Advances in Science and Technology* 56 (2008): 595–600.

[2] Manuel Kretzer and Dino Rossi, 'ShapeShift', *Leonardo* (2012): 480–481.

[3] 'Materia', accessed August 3, 2012, http://materia.nl/.

[4] 'Material ConneXion', accessed August 3, 2012, http://www.materialconnexion.com/.

[5] Blaine Brownell, *Transmaterial: A Catalog of Materials That Redefine Our Physical Environment*, 1st ed. (Princeton Architectural Press, 2005).

The first group of sensors is what is used for communication from users to devices. Although buttons, knobs and sliders are traditionally the most common way of providing input to microcontrollers, they require a deliberate action from the side of the user. Due to such required intentionality of providing input they are well suited for control of artificial systems. However, at the same time they also require extensive attention of users, and provide distraction from other performed by them activities. When more input is required, a non-functional button or know is not sufficient. In consequence building automation and security systems are often equipped with various kinds of control consoles, ranging from LCD and key panel, to touch-screen based. Web interfaces are also common, allowing a remote computer or a smartphone to control the building system.

In all the above cases, interfaces have the purpose of control, and don't facilitate or suggest interaction with a building or architectural space. As observed, control in such cases is perceived as hindrance rather than a natural process. With increasing number of controlled parameters of space, the ratio of users unable to perform the tasks at hand increases.

The second group of sensors are used to monitor the qualities of the environment and collecting feedback from the system itself. Among these are thermometers, light sensor, humidity meters, but also potentiometers and switches integrated in components to monitor their own state. Such sensors are usually hidden from the users, as no explicit input from users is expected.

Sensors employed in the studied prototypes are aimed at exploring new interaction modalities with interactive spaces, where interaction would be a natural and unobtrusive process. As investigated, a balance needs to be found between ambient and intentional interaction between users and architectural components. In this process the architectural space and environment may also serve as an intermediary agent in that interaction. Despite not having an explicit ability to actively interact, architectural space is transformed in effect of activities of users, components and external factors, and can be actively sensed by both users and building components.

b) Process

The Muscle Transports Prototype at Centre Georges Pompidou utilised three types of sensors in discs located on installation perimeter. There were passive infra-red body heat sensors, active infra-red proximity sensors, and touch sensors. Consequently, the installation could detect just remote presence of passers-by, individuals approaching closely and eventually a deliberate act of touching the disks. In this way various modes of engagement of users could be detected and the installation was reacting accordingly to the degree of sensed degree of participation. However, as observed, the affordance of sensor disks was not understandable to all users. Frequently users would attempt to turn the disks (similarly to steering wheels), which has resulted in damage to some of them.

In numerous following installations infra-red and ultrasonic proximity sensors has been the main choice for designers (among others in Muscle Tower, Muscle Space and other), as they allow general sensing of intentional and accidental input from users.

Other common sensors have been the pressure switches installed in the floor. The applications and specific sensors used varied, but performed similarly. In Muscle Space or Odyssey trigger mats were used. In protoDECK integrated Force Sensitive Resistors were used. In all cases this approach allowed estimation of where users are standing. As observed, accidental interactions were often followed by deliberate triggering of certain behaviours after users would learn from their own experience how installation behaves. More comprehensive sensing of presence of users were attempted by employment of video camera tracking and Kinect camera tracking, however, in these cases the requirement of a pc computer has proven to be too constraining.

A different kind of sensors has been used in the Ledworks project. The individual objects were equipped with accelerometers and gyroscopes. Users holding the objects in their hands, rotating or shaking them would trigger different colour changes.

In interactive environments Aeolus project; the rotary sensors were used in conjunction with small wind turbines. Users blowing into these turbines would trigger the inflation of corresponding wall components. Even though technically pressing a button would have had the same effect, the act of blowing, its affordance and resulting from it social interactions have delivered a different experience and influenced the interaction process.

In numerous projects various environmental sensors, such as temperature, humidity or barometric pressure were proposed and tested, however in none of the prototypes they were fully integrated. This might have been caused by the slow changes in these sensors' data, which were difficult to both test and present in the fast prototyping processes.

The last group of sensors used were sensors directly monitoring the performance of individual components. In this way linear potentiometers were embedded in linear actuators to provide feedback on their extension and allow exact positioning. Similarly, air pressure sensors were used in Cloud components to monitor internal pressure in cushions and prevent breaking them by over-inflation.

c) Evaluation

In traced projects various sensors have been tested. What has shown to be the main difficulty was the lack of specialised knowledge, time and materials to develop exact sensor needed for applications. Because of this available substitutes were often used. Attempts to self-engineer sensors have been educational for designers, but in many cases resulted in non-robust solutions. Especially in internal control feedback loops within components, such unreliability was unacceptable and resulted in destruction of components. "plug-and-play" sensors, such as those coming from Phidgets series[1] proved to accelerate the prototyping process.

1.6. Discussion

The section has provided an insight into the prototyping processes of interactive architectural components. It is established that prototyping is essential in development of innovative design solutions[2]. In studied cases prototyping started with delivering an elaborate taxonomy of forms and related material aspects to eventually provide an overview of internal processing, effecting and sensing sub-components. In the traced variety of prototyping techniques, it can be generally observed that design prototyping has often been a "quick and dirty" process, where decisions were often made in an ad-hoc manner, based on availability of techniques, and superficial knowledge.

[1] 'Phidgets Inc. - Unique and Easy to Use USB Interfaces', accessed August 3, 2012, http://www.phidgets.com/.

[2] Tom Kelley, 'Prototyping Is the Shorthand of Innovation' 12, no. 3, Design Management Journal (2001): 35–42.

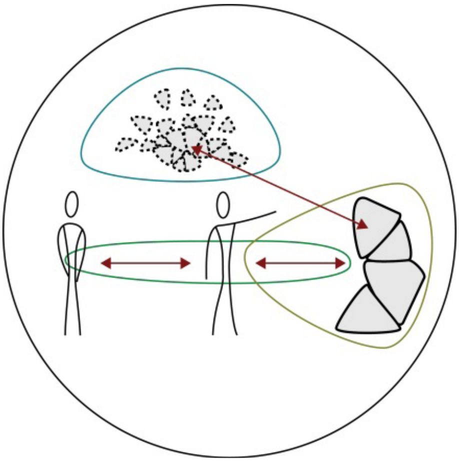


Fig.58. Ideogram: instruments for prototyping and communication.

However, when working solutions were established, a process of refinement would begin, when iteratively better solutions were found, at times replacing the initially assumed ones. What can be generally concluded, is that the component prototyping process benefits from availability of ready to use techniques and materials, permitting rapid testing of alternatives in the working setting. On the other hand, availability of technical solutions and knowledge of previously prototyped components has also shown to have influence on designs through inspiration.

This component prototyping process can clearly be improved in a multitude of ways. In this, knowledge and technical solution exchange between designers and projects is highly instrumental. Student designers have been extensively taken advantage of websites such as Instructables[1] or other online communities and for a (such as grasshopper or Arduino). However, a knowledge exchange platform focused on development of interactive architectural components would clearly further benefit this process.

Conclusions summary:

- Prototyped solutions provide complex technological challenges and require further, iterative improvement and development involving multidisciplinary experts.
- Integrated community of distributed systems "prototypers" is needed.
- Better knowledge exchange is essential
- Extensible libraries of components and their parts are needed

2. Forming iA networks

Challenges:

- iA is to be developed as an ecosystem of building components, people, spaces and virtual agents. All these entities need to form one complex network.
- An integrated approach to assembly, extension and reconfiguration of iA component networks is needed.
- Interoperability of heterogeneous components is a bottleneck in further development.

[1] 'Instructables - Make, How To, and DIY', accessed August 3, 2012, http://www.instructables.com/.

To this point the chapter has presented how various kinds of architectural components with embedded abilities to dynamically interact have been developed. The development of such components, however, is not the main challenge of interactive architecture design. It is means to an end, which end is formation of complex adaptive architectural spaces. In this respect, the process of assembling a network of architectural components, that form architectural spaces and that involve inhabitants in this process is the main task in the iA design and development process.

When looking at the differences among the interactive projects developed at Hyperbody over time, what can be observed is a shift of the topology in which the assembling of iA systems and communication among the various agents occurs. Initially, in all Muscle Projects it was a centralised topology, where all system's actuators and sensors were controlled from one point (although that one point contained a virtual multi-agent system). In the Leafs portal project, the system had a decentralised topology. Individual leafs had some autonomy, but they communicated through a coordinator holding the virtual model using protoSWARM and a variant of protoBASE. In the sCAPE project only locally communicating nodes constituted the project, where communication between nodes could occur only locally. In 751 city and reNDSM, entire, urban scale projects are set up in a distributed manner, where eventual global design ultimately emerge out of numerous sub-projects, which in turn are aggregations of flexibly defined building components and involve bottom up participation of user communities.

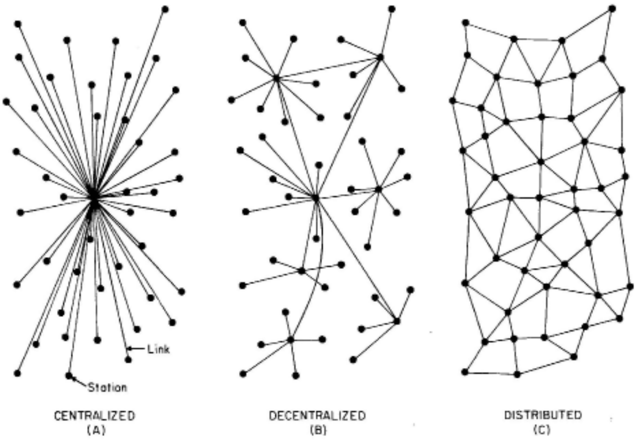


Fig.59. Centralized, decentralized and distributed network topologies[1]

Development of iA systems made up of autonomously operating component agents naturally leads to distribution of tasks and behaviours throughout such networks. A centralised network has many disadvantages. It is dependent on its core and when that core is not functioning the network fails. What's more, centralised networks are not scalable. There are always a limited number of connections that a network core can accept. Eventually, does not provide extensive flexibility, as every change to the network needs to be solved centrally. On the other hand, distributed networks are fully scalable, flexible and adaptive. A failure of one or more distributed network nodes does not result in failure of the entire network, on the contrary, in many applications other nodes can take over the functions of the ones that failed. For this reason connecting architectural agents into distributed networks is the most appropriate solution for architectural applications.

[1] Paul Baran, 'Introduction to Distributed Communications Networks', in *On Distributed Communications*, 1964.

This approach to operation of architectural system has few precedents. Traditionally centrally and top-down controlled systems are chosen for architectural applications. Therefore all standards and protocols existing in the field of architectural automation are meant to facilitate the centralised approach. Considering formation of distributed networks of architectural agents, new conventions and best practices need to be found and established. The preliminary level is creation of distributed end evolvable networks of architectural components. Eventually these networks need to be extended to include other iA agents, namely humans, virtual agents and architectural spaces. The main challenge is provision of conventions for development of such networks that would consolidate up to date efforts without compromising flexibility and imposing constraints.

2.1. Interconnecting building components

a) Outset

Architectural structures are built as assemblies of building components. These components need to be interconnected physically, exchange data, be provided with energy and eventually form working networks capable of higher-level functionality. Numerous problems are encountered when forming such networks and a balance needs to be sustained between uniformity of connections and heterogeneity of interconnected agents, extensibility, adaptability and evolvability of attained solutions.

Throughout case study projects a variety of manners for building up such networks have been investigated. The following points scrutinise the utilised means in respect to various aspects of connections between iA building components, point out constraints and identify further development opportunities.

b) Physical connections

Traditionally building components are connected in a permanent way, using mortar for stones and bricks, welding, rivets or permanent bolting for steel. Similarly, finishing often involves plastering or stucco and turns an assembly of components into a monolithic structure. This approach is clearly unsuitable for reconfigurable structures that are expected to be reassembled or reconfigured over their lifetime.

In executed prototypes every project included a different kind of connection between its components. Bolting connections were common but different connection types were used. Among those, differentiation could be made between node connections and edge connections.

o Node connections

Node connections varied based on the required properties of nodes, where the flexibility of the nodes was a decisive factor. Stiff odes were prototyped in the GEN project using bolts and bent steel plates connecting struts. This approach allowed full customization of the angles in the connection, but was restricted to connecting three struts. A wide range of other industry-validated static node connections exist. Cockpit and Sound Barrier node connections show some of many possible solutions.

In dynamic building structures the structural nodes often required flexibility. Chosen solutions varied, depending on the required degree of that flexibility. In the muscle tower project, steel spheres were used to connect struts with an axial, undersized bolt. Addition of rubber washers permitted strut deflection. In the Odyssey project greater flexibility was required. A solution of made of bolted together car-tire strips combined extensive flexibility with sufficient axial compression handling. In both cases disassembly was time consuming, but possible using common manual tools.

A different approach was taken for connecting the detachable Ledworks nodes in the sCAPE project. There magnetic connectors were used to directly connect nodes to each other. Consequently the nodes could be effortlessly and instantly attached or detached from each other. The disadvantage of the solution was the relatively low tensile force over such connection, limited to approximately 200N.

- Edge connections

In case of surface or volumetric components, which are required to deliver actual spatial boundaries in prototyped building structures, connections over the entire edge of the components are required. Similarly to node connections, edge connections may require varying flexibility. In the static components prototyped as part of the protoSPACE_4.0 pavilion (project led by Owen Slootweg, Chris Kievid, Christian Friedrich and Jelle Feringa) the problem of connecting solid EPS components has been very tangible. The prototyped solution included bolted tension rods inserted into the components. This solution provided required strength, but was not suitable for fast attachment and detachment of components. A quick secure and release system has been consequently proposed.

In author’s d|e|form project, hinging edge connections were needed to allow the folding of the structure. In the prototype the furniture hinges were used. By removing the hinge pin, the connection could be easily decoupled and re-attached manually.

In other prototypes, a variety of other connections has been prototyped, including magnetic or Velcro edges, zippers or custom made hinges. In all cases connections were highly customized and project specific. In some projects, such as the Flexscape, the connection has been the part of the actuation system.

This diversity results in a lack of compatibility between components designed for different projects. It has been acknowledged that in order to increase reusability of components and heterogeneity within designed iA system, conventions for connections need to be introduced in order to permit more configurations and to improve the ease of physical coupling and decoupling of components.

c) Data connections

The physical connection between building components is one way in which iA building components connects (arguably an iA building could exist even if its components would not be physically attached, yet their arrangement in space would create certain architectural qualities). From the point of view of developing iA buildings as distributed systems, connections between components allowing communication between them are most critical.

There are many technical possibilities for allowing microcontrollers embedded in building components to communicate with each other. Each of these possibilities has its own advantages and disadvantages. Some of such options have been tested in the course of development of prototypes and can be grouped in four generic categories: 1.) wired connections a) one-to-one connections (binary state signal, UART, RS-485) b) multiple elements on one bus (i2c, USB) d) ethernet e)internet-0 2.)wireless connections: a) Wi-Fi (IEEE 802.11) b) Bluetooth c) ZigBee d)digiMesh 3.)communication using visible light or sound. Many other possibilities exist, yet a detailed investigation into technical aspects of communication between such agents is beyond the scope of this dissertation as provided examples already clearly demonstrate the vast range of possibilities and typically encountered problems.

The simplest way to connect two devices to each other electrically is to transmit a binary on/off signal from one device to another. This is how a light switch operates. Similarly, signals of varying voltage can be sent, having the ability to control an explicit value. For example a typical volume knob in an old radio is connected to a potentiometer which controls the current voltage that goes through it and this voltage is used by the radio to determine the amount of sound amplification. This is also a way in which simple sensors and actuators can be connected to a microcontroller. This is also how a very simple bidirectional communication could be established between two microcontrollers using only two wires (an a common ground). The RS-232 serial binary protocol is in fact not much more than a slightly more sophisticated version of such communication. In its simplified version binary data, consisting of series of digital "on" and "offs" (binary high and low) is sent from one device to another in two directions on two separate cables. Simple convention (protocol) is established to regulate such communication and ensure mutual "understanding" of connected devices. Such connection has been used, among others, in the sCAPE LEDwork component connections. It is important to note, that physical making of this connection requires the crossing of a cable, i.e. one component's transmitting port (Tx) needs to connect directly to another component's receiving port (Rx) and vice versa.

In certain cases it might be beneficial to connect more than two elements to each other in one system, allowing one message to reach multiple recipients. This can be achieved using a bus connection, such as I2C. Communication over such system means that each element needs to have a unique ID and has to know IDs of its targets if a direct message is to be sent. Certain hierarchy is typically also introduced, as in such bus systems there is always a division between masters and slaves (although in I2C bus masters and slave roles can change dynamically). An example of application of i2c protocol is a protoDeck installation at Hyperbody Protospace by M. Hosale and M. Verde[1].

USB is a modern, more advanced bus protocol. Most computer peripherals use this protocol to communicate with the host computer. Programming of a USB port is however substantially more complex than i2c or RS232. It also requires dedicated hardware. For this reason integration of USB has not been experimented with on the microcontroller level. Additionally, host-client architecture of USB makes it a less interesting option for creation of non-hierarchical distributed networks. However, USB can be used when prototyping agent's behavior on a personal computer, where multiple devices can be connected to that computer (being a host) through one USB port. USB can also be used to emulate an i2c or RS232 connection from a PC to other peripherals.

Entities can also be connected together using an Ethernet network. Ethernet, comparing to protocols such as i2c or RS232 is a much more complex solution, allowing for several orders of magnitude faster data transfers and larger networks. Originally Ethernet used to connect multiple devices on one coaxial cable. Contemporary Ethernet uses radial network architecture and twisted pair cables running between network node devices and switches/ hubs. Ethernet Local Area Networks (LAN) are especially relevant due to ability to connect to many common devices and to the internet (internet being: inter communication between networks), as well as sending direct messages between devices (i.e. using the UDP protocol). Plug-and-play solutions exist that allow extension of easy to use microcontrollers (such as Arduino) with Ethernet connectivity.

Ethernet networks are widely becoming wireless with the adoption of IEEE 802.11 technology (Wi-Fi, standing for "wireless fidelity") that allows wireless connection of devices to Ethernet networks. Simlarly to wired Ethernet, even simple microcontrollers can be expanded with Wi-Fi connectivity at a relatively low cost. As Wi-Fi allows high-speed connection of a device to a wireless Ethernet network, Bluetooth allows connection of specific devices to one another over short distances at high speeds. It was originally conceived as a wireless alternative to the RS232 protocol. The only non-proprietary standard existing to date for distributed personal area networks is ZigBee, allowing for mesh network of nodes using cheaper and

[1] Marco Verde, 'Performative Design Processes for Architecture [P]a', in *Hyperbody: First Decade of Interactive Architecture*, ed. Kas Oosterhuis et al. (Heijningen: Jap Sam Books, 2012), 304–328.

less energy consuming hardware than Bluetooth or Wi-Fi, with building automation being one of the target markets and radio ranges of up to 1,5km in open space (30-70m for typical, low-cost ZigBee radios). ZigBee radios, however, don't allow a fully distributed mode of operation. Each ZigBee network has to contain a coordinator unit. Tests have also indicated that networks larger then 30-40 nodes begin to operate at very slow speeds. For this reason, a proprietary wireless network standard DigiMesh has been investigated. DigiMesh allows local wireless communication among nodes, without requiring a router. Extensive tests have proven DigiMesh to be scalable. Numerous other similar systems exist, such as e.g. MyriaNed however these systems have not been investigated.

The last category of communication exchange methods is much less efficient than solutions presented above, however it presents an interesting alternatives. Aside from established standards, experimental ways of communication may be established. An example may be using visible light to communicate between devices. An example here might be small student installations built in a workshop with Ruairi Glynn during the Interactive Environmnents Minor. Similarly to his installation Performative Ecologies, prototyped artificial "creatures" communicated with each other by flashing LED lights and light sensors. As much as this way of exchanging information is very slow and prone to data loss, it allows for communicated data to be instantly visible to humans. What's more, it also allows humans to interfere in such communication, by blocking the path of light, or even participate in it by flashing their own handheld light sources towards artificial agents. After the workshops, further experiments were made by students, including sending a low resolution image between two computers using a laser pointer. This encourages further experimentation with alternative media for exchanging data, that might be not justifiable from purely technological or performance oriented points of view, but which allow for creation of unprecedented user experiences and interactions in the physical world.

Listed solutions provide a range of robust alternatives for communication between network nodes. What's more, various communication protocols can be integrated in one system, operating as different layers of information exchange, where data can be forwarded or translated between layers by specific nodes.

d) Energy and matter flow connections

The physical connection and data connections provide primary foundations for creation of iA system. However, it the functioning of components often requires flow of various forms of energy and matter through components.

In studied examples, the electrical energy was required to power individual components and fuel their microcontrollers, sensors and effectors. In all cases electricity was provided from outside of the developed iA system. In the Muscle projects, compressed air was used as a different form of energy, specifically powering the Fluidic Muscles actuators. Among the studied projects, the Bubble Lounge was the first one to propose distribution of components and their autonomy. In that scenario, direct powering of elements appeared as an obstacle. If every component was to be connected to a central power source with a separate cable, the number of cables would destroy the appearance of the installation, become a significant expense and reduce the flexibility of the installation. Consequently, designed structure and preliminarily prototypes of components were designed to transfer energy and compressed air from one to another, theoretically requiring only one of the components to be connected to the external energy source. This approach has been taken over to following projects consisting of distributed components, such as LEDworks or d|e|form. In the first prototype of the LEDworks nodes, each node would contain a battery that would charge once the node was connected to another powered node. Detaching a node would allow its autonomous operation for up to several hours, until the node would be reconnected in a new place or the battery would run out, disabling node's operation. Notably, in the third version of the

LEDworks project developed by the student startup has reverted to powering every node directly from the electric line. This was mainly due to high costs of a battery, charging circuits and engineering problems with connectors. The d|e|form project focused on a simpler solution. There standard plugs were used to connect bower power and UART data between touching panels. The solution performs without noticeable problems, although electrical properties would need to be reconsidered when scaling the system up to above 10 panels.

In case of electrical connections between components, we clearly deal with the flow of energy, although this energy is carried by flowing electrons. In case of the flow of air, it is more clearly the flow of matter through components that permits their actuation. Consequently, further consideration can be given to other types of matter that would be desirable to flow through components to achieve various architectural affordances. In most buildings supply of fresh water is essential and removal of sewage and other forms of waste is equally essential. Fresh air needs to access inhabitants in buildings and CO2 needs to be removed. The same air often needs to be warmed, cooled or purified and its sound transmitting properties need to be in various ways influenced by architecture. Eventually also goods and people need to flow through buildings. Some of these challenges can be solved by providing through-component flow. Other, such as flow of people, need to be facilitated holistically by aggregations of components

e) protoNET as an evolving architectural networking platform

protoNET has been originally envisioned by the author in 2010 as a setup for a generic network of component nodes. It was a response to the fact that every supervised student project would either fail in building a prototype of locally networked nodes, or it would take an extensive amount of time of designers and the achieved result would be of low technical quality and robustness. It was observed that prototyping would progress significantly quicker and achieve significantly better results if students would start their prototyping work by modifying an already working technical solution. A generic form of such solution has been consequently called a prototype "template".

protoNET was proposed as such a template for setting up networks of component nodes. Such networks would need to be open, extensible, simple to use, low-bandwidth network, connecting autonomous sensor/actuator nodes spread throughout physical environment and concentrated on the developed project site. protoNET's objective was to provide a scalable and robust communication layer that would facilitate (ad-hoc) interconnection and cooperation of ubiquitous computing devices e.g. sensor nodes, experimental robots, components of buildings, control/information terminals, virtual nodes in a simulated environment and other. Emphasis was put on ease of use, flexibility, speed of deployment, and robustness. Data transfer speed and application of state-of-the art technological features were not main drivers behind the idea of protoNET. Each protoNET node was to be uniquely addressable, able to be queried for its sensor data, or with a request to execute an action, decide whether or not to respond to specific queries and itself query other nodes. protoNET was envisioned to operate as a wireless and wired network, it is not tied to any specific network architecture, carrier or protocol, although distributed network architecture was preferred. protoNET could consist of multiple sub-networks connected through gateways.

This initial idea has iteratively developed from stating with using Arduino FIO controllers boards with XBEE Series1 radio modules as controllers for building components. In this, a simple text message based protocol has been proposed for exchange of information between components. In pair with this idea, a more powerful Seeduino Mega microcontroller was used. That microcontroller also allowed using on-board UART connections to connect to adjacent components through two-wire cables. This solution has been applied in d|e|form in combination with shared power and data connections. Consequently the power connector type, the powering circuit and connectors to sensors and actuators were added to the description of protoNET.

Although the preliminary version of protoNET worked well in the d|e|form installation, scalability of this solution to larger systems proved questionable. Various radio standards, size constraints or feature requirements led to preference of designers for other controller platforms and communication methods. Various projects required different connectors for power or data. Aside from the technical aspects, the price and accessibility of other platforms proved advantageous for other projects. It has been in result acknowledged that protoNET should not attempt to enforce a specific "standard" for building component network, instead it has been beginning to evolve into an extensible practical knowledge resource base with hands-on information on how to build such networks. In respect to this, it has converged with the development of protoWIKI.

f) Evaluation

As shown, building components can be "connected" in multiple ways. Physical and data connections are the most fundamental connections that need to be defined for every project; however flow of energy and matter into and through the designed components also require attention. The way in which building components can connect and assemble larger structures determines possible affordances of these structures.

On one hand standardization of ways of connecting components can lead to more robust projects, where high-level functionality can be achieved and higher diversity of interconnectable components can be reached. On the other hand, such standardization can lead to uniformity and can constrain projects that require different solutions than in a given standard. Therefore instead of enforcing standards, open connection conventions can be established in order provide cohesion between gradual improvement of precedent solutions, and provision of new solutions.

2.2. Interconnecting things and people

a) Outset

Previous section has scrutinized various aspects of interconnecting building components into working systems. Systems of building components require close engagement of their users, without which buildings lose their *raison d'être*. A network of architectural components gives capacity for an iA system to adapt and deliver spatial qualities and affordances to its users "transforming everyday utilitarian space into an inter-activating network of spatial correlations"[1]. Interactions of such network of building components with people validate and drive development and evolution of iA sytems. Consequently iA buildings develop as ecosystems of building components and people (further extension of these ecosystems by virtual agents will be discussed in next section).

Building components can be interconnected by especially designed connection interfaces, whether physical or digital. In case of people, interactions have to occur through interaction modalities allowed by human body and brain and therefore are significantly more complex and difficult to predict. Studied projects have established first steps into that direction, allowing identifying threats and opportunities.

[1] Nimish Biloria, 'Interactive Morphologies: An Investigation into Integrated Nodal Networks and Embedded Computation Processes for Developing Real-time Responsive Spatial Systems', *Frontiers of Architectural Research* 1, no. 3 (September 2012): 259–271.

Building components discussed in section 1. included examples of sensors and effectors that can be used for communication between a person and an individual components. Discussed interactions that these interfaces afforded can be classified as ranging from direct and explicit feedback when the user consciously provides information she chooses to provide, to ambient when the user does not choose to provide information and may not be aware of information being collected. In reality interactions between users and interactive components lie between the two extreme situations. On one hand if repeated over time, human actions of providing information to interactive devices become automatic and provided less consciously. On the other hand, over time users learn how devices gather data from them and make conscious use of that knowledge, for example noticing where the infra-red sensor triggering the door to open is located and waving their hand in front of it if the door fails to open.

The interaction between users and building components can be compared to interpersonal interactions. We gather information about each other in a non-direct way using all our senses. We judge each other's looks, body language, clothing, scent, can observe each other's activities and through know references develop an idea on who the person is. In some cultures the amount of openly provided information about oneself can be limited, e.g. by clothing such as burkas. In other cultural contexts individuals may want to share additional information, e.g. communicating their social status through clothing and jewelry, publically pronouncing their opinions or carrying banners or clothes with slogans or symbols. Once direct interaction between humans is established, the communication becomes more conscious and proceeds typically using several modalities, such as spoken language, face expression and body language. Culturally we have also developed media as intermediary means of communication[1]. We can use written text, audio or video transmission to communicate without direct contact. Digital media and social media provide even greater possibilities for social interaction.

Same possibilities as faced when communicating among humans can be employed to communicate between humans and building components. In popular fiction (e.g. Space "Odyssey 2001" or "I am Robot" movies) interaction intelligent architectural environments interacting with humans are attributed with some human traits, such as generated voice or advanced vision. In these cases, however, there is one intelligent artificial agent controlling the building, whereas in the proposed approach many intelligent agents collectively constitute a building. What's more applying human attributes to non-living objects for communication purposes seems inappropriate in architectural applications seems interactions there are expected to be more ambient and individual devices closely integrated with one another.

Consequently, new interaction models need to be found for interaction between human and building components, eventually building up new societal and cultural conventions.

b) Process

For the purpose of studying interactions, in Muscle Transports installation, every sensor disk located on installation perimeter can be treated as an independent component, despite the fact that all disks were connected to one central computer. Muscle Trans ports has sets an example of an interaction model that using simple means provides a gradual transition from ambient to conscious interaction. Each disks' PIR and IR-distance sensors trigger contraction of surrounding muscles when detecting passers-by. Passers-by may not be aware of triggering the activity of the structure, but after passing the installation several times, they are likely to notice the effect their presence has on the kinetic structure. Once deciding to verify this observation, they would stop, and the response of the structure would be intensified as they

[1] Marshall McLuhan, *Understanding Media*, 2nd ed. (Routledge, 2001).

approach closer and stay longer in front of the sensor disk. Eventually noticing and touching the force sensitive resistor located in the middle of the disk would trigger the most dynamic contraction and expansion cycle of the neighboring actuators.

Other Muscle projects included similar interfaces, where sensed presence of a person in front of a particular component or act of touching an FSR sensor would result in pre-defined sequence of installation's actions. In some projects other forms of gathering input from people were used, such as floor trigger mats in Muscle Space or Odyssey, or accelerometers embedded in sCAPE ledwork devices. All employed sensors allowed different degrees of interaction "consciousness" in users and helped users to learn the interaction through exploration.

The connection between users and building components occurred through a learning feedback loop, where learning happened on the user side. The users would learn how his activities are sensed and mapped to specific behaviors of architectural structures. In this process, the behavior of the structure was the response. With few exceptions, no other feedback mechanisms were embedded in the interfaces. Users would know that stepping on a senor or approaching a component works only by experiencing the triggered response. This stands in contrast to other commonly encountered product or building automation interfaces, where interface feedback is often separated from the affected reaction.

In Muscle and Interactive environments minor projects prototyped interfaces were limited in their extent due to the time and skills of designers and prototyping material available at hand. Additionally, designers were explicitly guided to avoid screen-based interfaces in order to focus on design and prototyping of interfaces directly integrated in designed structures.

In Hyperbody MSc1 courses students had more design freedom in respect to interfaces and fewer constraints related to prototyping requirements. In many of these projects designers chose for inclusion of web-based interfaces for projects and integration with social network platforms. Among others the Spacebook project proposed an interface where users would input parameters of space required and the system would automatically generate the needed spatial configuration. In the Architecture Republic project its designer Chao Wang proposed an online order system where users could parametrically define requirements for needed space, and a module would be fabricated and automatically delivered to the requested location.

What has been observed is that when not given incentives to design and prototype iA systems as distributed, designers tended to choose user-control interfaces over interaction-based interfaces and centralized interaction with buildings (using a console or a web interface) over spatially distributed one. Among reasons mentioned to justify these decisions was the familiarity of control systems and it being easier to comprehend and predict development in centralized systems.

However the growing popularity of smartphones and other mobile devices with internet access has allowed merging the centralized interface approach with spatial distribution of interactions. Smartphones provide a new interface to the building components through a web gateway service. An example can be the application of the protoTAG system. In the reNDSM project smartphones were used by users/designers to identify scanned system components incorporated into the project system and add information to them. On the other hand, numerous projects used smarthpones as remote controllers for specific building components and designed extensive smartphone applications where communities of participants could together drive the development and transformation of the architectural project. Author's proposals extend this vision beyond architecture to the urban scale[1].

[1] Tomasz Jaskiewicz, 'Complex Multiplayer Urban Design System – Concept and Case Studies', ed. M. Forsell et al., *Euro-Par 2009 – Parallel Processing Workshops* 6043/2010, Lecture Notes in Computer Science (2009).

c) Evaluation

The formula for human-component interface design in interactive architecture remains open and highly depends on the project context and reception of interfaces by user groups. Following the parsimony design principle it can be stated that the simpler the proposed interface to fulfill the given task, the better is its design. On the other hand the "task" of an iA building is a moving point, therefore iA interfaces need to remain generic for future, not yet known interactions and affordances of iA structures.

As many of iA functionalities are expected to be performed ambiently in the background, indirect interaction between users and iA components is essential to be maintained. Users quickly develop an expectation that an iA building should continuously adjust itself to their activities, or event anticipate them. On the other hand in this process the affordances delivered by buildings are bound to influence needs and desires of its users. In that respect provision of feedback through actual building transformation is a promising direction for shaping such reciprocal processes.

At the same time, human-iA interactions may go beyond the direct communication. Mobile, often web-enabled, devices can serve as intermediaries in those interactions. They can identify the user and provide additional information to the building and the other way, provide users with information about specific building components and interfaces to control or interact with them.

2.3. Interconnecting physical and virtual

a) Outset

Previous two sections provided an overview of the prototyped solutions for forming and connecting building components, and for actualizing the connections between these components and users. As discussed in previous chapters, studied iA systems also consist of virtual agents. In the initial design process stages these agents dominate developed systems. Over time physical agents grow in numbers and virtual agents may (but don't have to) reduce their numbers. Nevertheless, they remain active in the designed iA systems. Therefore active connections between virtual and physical agents need to be supported in prototyped systems.

Initially there has been a strong methodological division at Hyperbody between the projects focused on prototyping, namely the Muscle projects, and projects where only conceptual design was the end-deliverable. In the Muscle projects the full-scale prototype building has always been initiated early in the design process, based on initial virtual sketches and simulations. The virtual system would then iteratively develop in parallel with the physical prototype. In this method the virtual systems would be reduced to control systems for the physical installations. In other projects the entire project would develop virtually, first as a sketch, eventually as a digital system of virtual agents. The complexity of the virtual system would grow over time and reach a very high degree of complexity, yet without any form of empirical verification.

Each approach had its advantages, as discussed in chapter IV. Ultimately, for complex projects the challenge lays in convergence of the two approaches and seamlessly integrating physically prototyped systems into the highly complex, virtually prototyped ones. The key factor in achieving this is the interconnection between physical and virtual systems and agents, which need to form one working architectural ecosystem[1].

[1] Chris Speed, 'An Internet of Things That Do Not Exist', *Interactions* 18, no. 3 (May 2011): 18–21.

b) Process

Starting with Muscle Trans-ports, the muscle projects have been based on the same system model of interconnection of its virtual and physical agents. Virtual agents simulating the installation were created and deployed in the Virtools platform. Two special virtual entities were added to the virtual system. One entity (“input agent”) dealt with collecting all sorts of data input, the other was controlling output (“output agent”). In this setup all virtual project components would communicate with these two entities. The input agent used midi protocol and in later projects also phidget board interface to connect to sensors embedded in physical components. The output agent connected through a serial interface to a relay board, which in turn controlled centrally located air valves individually controlling inflation and deflation of all actuators. This setup provided a mix of distributed multi-agent operation of virtual systems with centralized infrastructure. The solution of the infrastructure was sufficient for muscle projects, but provided very limited scalability and therefore was not adequate for further project development into more complex and dynamic systems.

An solution to extend the centralized approach was conceptually proposed in the iPortals project. There the projects involved very short conceptualizing stage and development of prototypes went hand-in-hand with virtually designing their behavior. There every installation consisted of one (Curtain, Pods, Jealous Portal) or several (Leaves Portal) PC control units. Each such PC unit would host the virtual system of either an individual interacting agent or several agents, built using MAX|MSP software platform, which did not explicitly support creation of virtual multi-agent systems. The control infrastructure was similar to Muscle projects, namely every PC unit would connect directly to installation’s sensors and effectors.

In consequence of this setup, very early in the design process the distinction between virtual and physical agents dissolved. Every PC computer became treated as a physical agent with a certain behavior, connected to one or more installation components. As the prototyping progressed, in Leafs and Curtain projects the PC unit would become integrated in the installation components. In the Leafs Portal project this integration was direct, each leaf had a laptop computer embedded in its base. In the curtain project it was indirect; the pc controller was embedded in the base part of the installation and remotely actuated the kinetic panels, while gathering sensor input from proximity sensors located in the base. In this way, even though the curtain consisted of a number of discrete components, it came to be perceived as an integral whole; only one interactive agent, not a system of such.

The additional novelty in the iPortals approach was the envisioned meta-layer of connecting installations. The partly implemented idea involved connecting all PC-controllers into one network, where individual installations would be part of a greater system of interactive space-defining objects.

Consequently the observed development process involved merging and integration of virtual agents into the physical ones, while the further (unfinished due to the faculty of Architecture fire) integration of portal projects would have required introduction of new virtual agents, and a likely persistence of their virtual operation throughout project lifetime. Possible applications of such approach can be observed in numerous projects from Hyperbody MSc1 design studios. There virtual agents include building spaces, activity groups, or events, which virtually exist next to the actual systems of people and building components, are directly informed by the physical system and provide instant affect on people and building components, despite not having an explicit physical embodiment.

Later projects from the Interactive Environments studio and Hyperbody MSc1 have involved various sorts of actual-virtual connection due to various development processes. With the increasing number of components and increasing complexity of projects, the ad-hoc and centralized solutions have become a bottleneck in project development. Additionally technical aspects of realizing that connection have been reported as frustrating and time-consuming for designers. In response, the d|e|form project involved a solution for scalable connectivity between virtual and physical agents.

In the d|e|form setup, a project can consist of any number of virtual multi-agent subsystems. Each such subsystem includes an input-output agent which connects wirelessly (XBee series 1 DigiMesh radio unit) to the network of embedded physical agents. If the physical agent is in-range, direct communication is possible. At the same time embedded agents can either wirelessly or using local cable connections (UART) communicate among each other. On the larger scale, virtual-sub systems can connect to each other through internet. Embedded agents sub-networks can similarly include internet gateways allowing long distance connectivity. The setup has been partially tested in the first d|e|form prototype and has proven to be robust. However, it has also presented itself with multiple layers of complexity in respect to continuous operation in realized projects and therefore requires further development, mainly in respect to data sharing conventions and technical reliability.

c) Evaluation

The development of applied techniques for interconnection between physical and virtual iA system agents show a consistent tendency towards diversification and distribution of connections on one hand, and establishment of conventions for ease of use and reusability of technical solutions on the other. Traced projects have demonstrated that little can the types of agents and their evolution in projects cannot be fully anticipated, but certain patterns do appear. Among such patterns is pairing of corresponding virtual and physical agents (i.e. virtual agents preceding or extending the functionality of physical ones), as well as a emergence of virtual agents that correspond to a multitude of heterogeneous physical agents (i.e. virtual agents organizing the functionality of sets of physical ones, e.g. “event” or “place” agent).

The development of the system for the d|e|form project can be seen as the first step on extending the protoKIT with instruments for flexible and easy to use prototyping of distributed virtual-actual connections between project agents and therefore significantly contributing to further development and increase of complexity of iA systems.

2.4. Discussion

The section individually looked at challenges related to interconnecting physical iA building components to each other, to their human users and to virtual iA agents. It has been observed that in all aspects of the investigated realization processes the diversity of solutions is very high and certain projects may require custom project specific solutions.

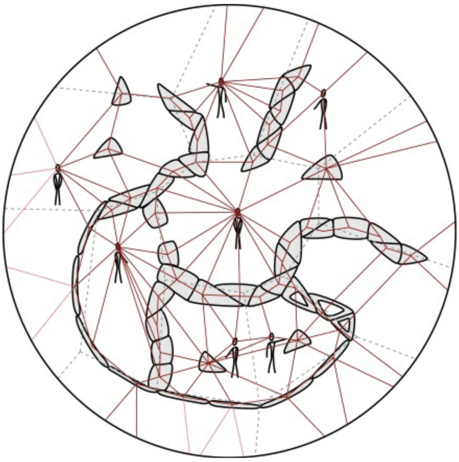


Fig.60. Ideogram: complex network of interconnected heterogeneous building actors.

In the process of adding agents to the iA systems with physical material properties and distinct spatial location, providing the possibility for these objects to form a network, communicate and interact with one another becomes a substantial challenge. In purely virtual systems the communication between agents is not affected by their location and is easily monitored and controlled. In physically distributed systems, various communication channels can coexist and operate across different media. Also monitoring and control becomes significantly more difficult. Yet most importantly, physical components are more unpredictable in their performance and behavior than virtual ones, whose behavior and properties are strictly and unambiguously defined. What's more, physical components can be influenced by factors coming from outside of the system and not accounted for by system designers (ranging from atmospheric conditions to vandalism).

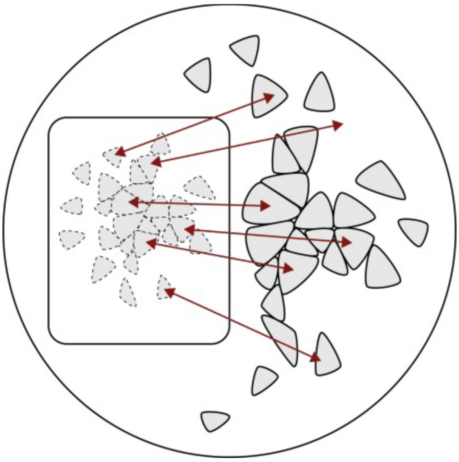


Fig.61. Ideogram: virtual-physical system integration

Such unexpected performance of individual system components becomes only increased through aggregation of components and complexification of information exchange and interactions between them. At the same time it is the unpredictability and emergence that permit evolution, achieving which is a fundamental aim of interactive architectural systems. Therefore strategies need to be found to channel the emergent and unpredictable into system improvement and adaptation instead of its destruction or malfunctioning.

The described techniques for realising components and interactions between them have focused on the technical aspects of providing necessary capabilities in iA systems for material properties and communication required for system development and evolution to be supported. The initial steps have sufficed to construct initially working prototypes. However, it is acknowledged that further increase of complexity of developed systems and their embedding in real-world situations will require further improvements. For this sharing of knowledge, including establishing conventions for all aspects of iA agent connections need to be continuously enhanced, updated, shared and improved in a larger community of iA designers and engineers.

Conclusions summary:

- Emergent properties of distributed networks of complexly interacting components can't be predicted.
- Unpredictability and lack of control need to be turned from risk to advantage.
- Open protocols and conventions for connections need to be established across designer communities and projects.

3. Realising interactions

Challenges:

- Active involvement of user feedback in the design process.
- Active involvement of material and performance feedback in the design process.
- Provision of means for developing alternatives to linear interaction design scenarios.
- Development of methods of engaging users in the design process.
- Creation of complex interactive iA ecosystems with ability to develop and evolve.

Chapters IV and V have dealt with tracing the virtual development of complex architectural systems and the parallel development of design environments and instruments. Eventually, in the current chapter, focus has been given to strategies for direct involvement of physical content and context in development processes of architectural systems and realisation of iA systems in such material context.

The two precedent sections consecutively deal with the challenges of materialisation and subsequent interconnection of architectural system components. The current section attempts to answer the question of how this materialisation and interconnection can be employed to design desirable interactions, perpetuating the development and evolution of iA systems? The current section may at first appear as partly overlapping with section 2. However, the critical difference is that section 2. focuses on the emergence of the network of iA components, while section 3. investigates how interactions are developed within such network, or in other words, what the edges of the network actually mean and how they build up interactions.

Approach towards architectural interactions prototyped in case study research projects, divides these projects in two groups. The first group are the older projects following the Muscle NSA that were based on Fluidic Muscle actuators and the associated centralised control system. In these projects development of one-to-one interactions between behaviour of the whole building and one user or a homogenously and integrally approached small user group have been the main focus. The second group are later projects starting with the iPortals projects and continuing in the projects executed within the Interactive Environments

Minor course. In those projects more focus has been put on development of distributed interactions involving many components and/or several installations and many individual users forming larger heterogeneous user groups.

The challenge set for the current section is to investigate how design of actual building-user interactions can be advanced and improved through involving users in the design, prototyping and operation of interactive buildings. Because of the nature of the interactive process, design of such interactions inherently involves change in both; users and the building agents alike. What's more, interactions among users and among building components are an equal point of concern. The ultimate goal of the iA systems is to create conditions for interactions that through mutually influencing feedback loops promote mutual growth in pair with development and evolution of architectural systems. New strategies and methods need to be found in order to successfully reach this goal.

An interaction between two agents can be considered as a basic "building block" for designing complex interactive systems. By definition[1], interaction involves engagement and transformation of all agents that interact with each other. This stands in contrast to some popular preconceptions about interaction in architecture, where architecture is expected to be transformed in response to desires of its users, but its transforming effect on the users is not taken into account.

Many design methods exist that assist and guide design of interaction, many of which stem from user-centred design approaches[2] (that are common in industrial design and relatively uncommon in architecture). Among them, methods such as contextual enquiry[3] or cultural probes[4] focus distinctively on developing an understanding of users, their problems and context of these problems. These initial design phase methods are followed by methods such as Wizard of Oz[5] and other forms of testing and evaluation of performance of proposed design solutions against initially identified problems.

However, architectural systems deal with different sets of problems than typical product design, and many of the established methods require adjustment or can be constraining to the design output. Specifically interactive architecture is inherently multi- and trans-functional. That means that it is not expected to respond to well-defined problems of users, but instead it needs to continuously identify these problems and adapt itself to deliver new solutions. In respect to this, established user enquiry methods are not suitable to be embedded directly in the interactive systems. Conversely, interactive architecture is projected to act pro-actively, meaning that it can actively influence or even generate new needs and desires of its users. Consequently user feedback can be used as a guideline and indicator, but not the only driver of iA development and subsequent evolution of iA systems. Eventually, co-creation is also expected to play a critical role in realised projects[6]; however no explicit techniques exist for setting up co-created systems without imposing top-down control on these systems.

[1] Haque, 'Architecture, Interaction, Systems'.

[2] Bruce Hanington and Bella Martin, *Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions* (Rockport Publishers, 2012).

[3] Douglas Schuler and Aki Namioka, 'Contextual Inquiry: A Participatory Technique for System Design', in *Participatory Design: Principles and Practices* (Routledge, 1993).1993

[4] Kate Herd et al., 'Development of a Design Probe to Reveal Customer Touch Points in the Sale of Mass Customised Products.', *Design Principles and Practices*, 2009.

[5] S. Dow et al., 'Wizard of Oz Support Throughout an Iterative Design Process', *IEEE Pervasive Computing* 4, no. 4 (December 2005): 18 – 26

[6] Elizabeth B.-N. Sanders and Pieter Jan Stappers, 'Co-creation and the New Landscapes of Design', *CoDesign* 4, no. 1 (2008): 5–18.

Accordingly, no specific methods have been initially assumed for designing iA interactions. Many techniques such as Wizard of Oz, storyboarding or design personas were employed at various design phases, however each of these techniques encountered limitations. Ultimately, the frequent design-prototyping-testing iterations were chosen as preferred design practice for its high flexibility and possibility to swap design methods and techniques between iterations. The following accounts of developing interactions in projects bring novel insights into the nature and processes that guide formation of interactions between users and dynamic architectural systems and grow in consideration from direct, 1:1 interactions to many:many interactions where aggregations of building components and groups of users are accounted for. The purpose of this section is to trace the processes in which interactions were translated from design ideas to working processes. This tracing is aimed at developing a deeper understanding of how these processes can be further improved and advanced in order to reach higher degree adaptability in pair with high numbers and heterogeneity of components in realised iA systems.

3.1. 1:1 Interactions

a) Outset

In order for any of the 1:1 interactions to exist, a feedback loop is required between interacting agents across a connection consisting of one or more communication channels[1], and inducing mutual transformation of the interacting agents. The nature of such feedback loop requires to be designed.

In the formation of interactions, it is critical to specify what information exchange constitutes the feedback loop, or approaching the issue from a different angle; what transactions agents perform among each other. In hierarchically structured decision systems there are no interactions, as decisions are made on top of the hierarchy and propagated downwards, while all information necessary to make these decisions is sent in the opposite direction. In interactive systems interacting with each other agents make autonomous decisions guiding their own action (where every action induces some form of self-transformation), but can mutually affect each other's decisions. That mutual exchange of information leading to reciprocal transformation can be referred to as transaction.

Consequently, the initial rudimentary question that can be asked at the outset of any iA project is how to sustain continuous transactions between various kinds of components in an iA system. The challenge at hand can be split in three groups of distinct interactions. The first group deals with interactions between users and artificial building components, and is characterised by disparity between the intelligence of users and simple behaviours attainable in artificial components. It can be generally established that current interfaces used in home and building automation are inadequate for iA purposes. "Future homes could be able to sense situations, react appropriately and inform the user. Such aware systems will need to have a mechanism to determine when and how to interrupt the user."[2] New cultural and psychological conventions need to be established for such mechanisms to operate. The second group involves development of interactive processes among artificial components. Here again, no standards exist for distributed communication among building elements. The third group concerns development of interactions among users in iA systems.

[1] Shannon and Weaver, *The Mathematical Theory of Communication*.

[2] M.H. Vastenburg, D.V. Keyson, and H. de Ridder, 'Interrupting People at Home', in *2004 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, 2004, 59 – 64 vol.1.

b) user:component interactions development

As Donald Norman writes "Taming technology requires a partnership between the designers and those of us who use it. The designers must provide, effective communication, and a learnable sociable interaction. We who use the results must be willing to take the time to learn the principles and underlying structure, to master the necessary skills. We (users ibid.) are in partnership with designers."[1]

Initial interactions designed and prototyped at Hyperbody were built around simple interaction scenarios involving the installation as a whole or one of its components reacting in a predefined way to a set of generically specified actions of a user mediated through embedded sensors. The scenarios would not differentiate between individuals or in any way be influenced by the history of earlier interactions. The reaction scenarios were easy were formulated or easily translatable to a short series of "if... then..." statements, such as "if you sense a person passing by, contract the connected actuators for 2 seconds", "if you sense a person touching you, contract the connected actuators for 5 seconds, release them and start contracting again for as long as the touch is sensed". In these scenarios it was assumed that the user would learn how the system operates through exploration and adapt her activities to installation's response, either by taking advantage of newly created spatial affordances, or provoking further transformation. Additionally, most interactions designed in Muscle projects followed the agenda of creating experiences where installation would be perceived by its users as a living artificial being forming an inhabitable space. Such perception showed to strengthen the urge of users to involve themselves in a long-lasting interaction with the space, instead of expecting it as a controlled artefact which responds to predefined commands.

Gradually, designers were encouraged to develop more detailed concepts for the development of interaction between individual users and dynamic architectural spaces. That was achieved through linear interaction storyboards. An example of such storyboarded interaction can be the operation of the Muscle Space project. The design intention was to encourage movement of passers-by, while not enforcing it, and adapting the behaviour to users who decide to stay. The walls of the installation would condition the user's ability to pass through the designed space. The installation would sense the presence of a passer-by in each of its segments and in response contract the space of that segment, while opening up the adjacent ones, inclining the user to move forward or step back. Lack of response of the user would again reopen the segment she was in; showing that staying there was possible, despite not being encouraged. The visual pattern projected on walls enhanced the used emergent graphical patterns to increase the overall impression of being guided through the passage. Although no explicit learning was involved in installation's behaviour, the top-down narrative of the behaviour included splits based on user actions, thus allowing users to make a choice at any point of the interactive process, by continuing to move forward, stop or go back. Such choice would result in the installation switching to a different behavioural pattern.

In other projects such as the Odyssey (but also e.g. Muscle Bamboostic), component behaviour randomization was added to increase experienced agency of created spaces. Through the simple means of allowing random parameters to partly influence the change in installation behaviour, the experienced linearity and predictability of that behaviour became broken. In the Odyssey project it was manifested by semi-random waves of movement in the floor and walls of the structure, radiating from the user steps.

Coherently through many conceptual design descriptions of traced projects, designers described that the aim of linear narrative-breaking installation behaviours was to encourage users to change their typical habits an engage themselves in playful explorations of

[1] Donald A. Norman, *Living with Complexity* (The MIT Press, 2010).

prototyped environments, seeking new possibilities for socialising with other users or exploring new affordances that designed spaces provided. In many such cases, however, the user engagement was short lived. Although users reported generally, that the experience was positive and surprising, they would also soon assume that they have seen what there is to be seen and leave the space. This kind of reactions can be directly attributed to the lack of development in the behaviour model, and its consequent predictability over longer time.

Projects such as sCAPE or Flox from the Interactive Environments Minor developed models of user interactions that were more engaging on the long-term. In both projects any user was encouraged to detach a component from the structure and move it to a different place. The light change of the component would depend on the context it had been detached from, the manner in which it had been moved and where it was reattached. Consequently, the history of the interaction would matter for the momentary behaviour of each of the nodes and lead to infinite possibilities of non-random, but also not fully predictable interactions.

All interactions discussed to the point show relatively low degree of interactivity, as user actions had limited and short-lived effect on the long term behaviour of the installation. No learning of components had been developed, as creation of intelligent agents was beyond capabilities of student designers and timeframe of projects. An exception was the GEN project, where a virtual “kudo” currency was introduced to all components. Whenever the component would detect it is in use, its kudo count would increase, and the spatial configuration (beding or level of inflation) would be recorded as giving income. In this way, most usable configurations were remembered and if kudo flow would stop the next remembered as successful configuration would be assumed. If kudo flow would not be generated in a longer period, random configurations would be assumed in search for a new attractive spatial affordance. The “kudo count” of each component and association of kudo gain and loss with individual effector values provided a basic form of memory for each of the components. This memory was subsequently envisioned to influence actions of components. In this manner a basic form of component learning had been achieved and could be further built upon in other projects.

The d|e|form prototype utilised a system similar to the “kudo” economy. This system, however, was judged as not sufficient to solely support long-term development of the installation, where components may need to alter their functionality entirely. A user-driven component adaptation was therefore introduced as another form of interaction between users and components. In that case users were envisioned to be equipped with easy to use instruments to directly reprogram the behaviour of selected components, to fit better in the changing context of other components, spatial transformations and new needs and requirements. Proposed scenarios provoked many questions, such as how to encourage users to update the component behaviour? How to prevent errors and malicious reprogramming? How to decide and regulate who is permitted to reprogram components? To answer these questions more extensive prototyping in the realistic social context is required.

c) component:component interactions development

In early prototyped projects, the interactions among prototyped components were limited. All Muscle projects were controlled centrally by a PC unit. In those cases, direct interactions between components did not exist. Instead a central virtual agent would collect all sensor data and control the effectors. In that setup the installation operated as a singular interacting agent.

The original vision of the Trans-ports project envisioned that when multiple Trans-ports installations would have been built, they would communicate with other installations scattered around the globe and allow their users to share experiences, emotions and communicate with each other with the use of ambient spatial qualities. The iPortals project built upon that idea by conceptually proposing the four portal installations to form one architectural environment, where interactive portals would complement each other's affordances and cooperate in defining an integral architectural space.

In project of the Interactive environments minor, the focus has been put on identifying autonomous components on the smaller scale. Whereas some projects continued the singular-behaviour approach, some, such as sCAPE, GEN, or Blox followed the path where every component of the installation had its own behaviour and the global behaviour. In those cases interactions between components formed the foundation for installation behaviour and had to be developed in parallel with the interaction scenarios and other interactions in the iA system. In these projects, the data connection between components (described earlier in point 2.1.c), became the main bottleneck of the realisation of the installations. The d|e|form project provided a flexible set of conventions for interactive transactions between installation components. Additionally it also systematized the interactions between physical and virtual agents, allowing direct transactions between any pair of virtual and physical agents. Interactions between components were initially developed and tested in the 1:1 mode, however, in order to develop a complete experience for the entire installation a step towards design of many:many interactions had to be taken early on in the behaviour design process and will be discussed further in the next point.

d) user:user interactions development

In parallel to the increasing number of autonomus interacting with each other components included in the designed projects, interactions involving more users were developed. Consequently, interactions among users started to play a role in the designed systems.

Design briefs for the interactive environments minor always included a description of the inter-personal interactions that the installation should enhance, and designers identified inter-personal interactions that were not desirable. Most projects promoted social interactions including stimulating random encounters, providing shared reference experiences for discussion and socialising or giving users a collaborative goal in various forms of play.

Ultimately the transformation and enhancement of 1:1 interactions among users became the goal for many of the installations.

e) Evaluation

In the discussed user-component interactions, the development of interactions has been limited not by the interface itself, but by the complexity of component's behaviour. It has been shown, that user engagement can be achieved with the use of very simple interfaces comprising of as little as a button and a light source. However that engagement typically takes the form of control of the component, not an interaction with it. In further discussed projects interfaces have been more elaborate; however the initially designed interactions have been often reduced to simple, linearly defined response patterns.

Similarly to user:component interactions, only rudimentary data-connections were needed to develop complex interactions among components. Also there, the actual difficulty lied in programming the components with behaviours allowing development of interactions beyond simple reactions. Inclusion of learning agents allowed significant improvement of the complexity of interactions and increase in the degree of interactivity creating more engaging interactions and their perpetual improvement. Reliable development of such learning component agents proved however highly time consuming and difficult to designers. What's more, also this approach has shown its limits, as conceptualised learning agents were always constrained to predesigned boundaries of their behaviour model. The ability given to users to directly modify of building components by users proved to be solution for such situation, allowing the behaviours and physical features of components to be developed in a bottom-up fashion beyond constrains imposed on them by designers. However, giving users full control over components reduces again the degree of interactivity, and presents a risk of iA systems being developed back into controlled systems, as well as giving room to potentially malicious modifications by users.

The user:component interaction has drawn the most attention in developed projects. This might be attributed to the fact, that one user and one component may be seen as the simplest interactive architectural system. However, ultimately, interactions among components and among users have shown to be equally important and difficult to design.

Consequently, it the sift of focus from 1:1 interactions to many:many interactions becomes a necessity. Development of interactions of iA systems as wholes can be obtained through multi-component and multi-user collaboration and interaction, where learning and adaptation can emerge out of discrete simple interactions of individual components and user engagement and development of new cultural and social models emerges out of interactions of many individuals between each other. While the global development and evolution of the iA system as a whole occurs in heterogeneous ecosystems of humans and building components, both virtual and physical.

3.2. many:many interactions

a) Outset

As discussed in previous point, direct interactions between iA system agents can be seen as basic “building blocks” of largely complex interactive processes of iA systems, driving the development and evolution of iA systems as wholes.

Basic 1:1 interactions can be designed using established design techniques, resulting in a limited number of linear interaction scenarios, in which transactions between system components follow predictable patterns. Many:many interactions, however, present a degree of complexity where scenarios can only be used as vaguely guessed predictions about certain future states of the system. In that way, an interaction scenario can only be seen as a one-time projection into near future approximating one possible state or sequence of states of the designed iA system.

As a result, more open design strategies and design evaluation methods need to be found for developing many-to-many interactions and largely complex adaptability in iA systems. In such approach, prototyping of networks of interactive components may require a different approach than centralised interaction prototyping. New methods, techniques and instruments are needed for accelerating iterations of many-to-many interactions prototyping and iterative evaluation of design results.

b) protoDECK

protoDECK consists of 168 active floor tiles, where each tile is embedded with a force sensor, RGB led unit and a microcontroller. The installation has been originally intended as a direct physical extension of any experimental virtual system, allowing multi-modal interaction of many users with that system. In this setup each tile’s microcontroller would relay sensed events to the central computer hosting the virtual system and execute state changes of the RGB lights, based on commands received from that computer. That approach promised protoDECK to become a powerful and flexible device for prototyping multi-user engagement in interactive architectural spaces. However, due to centralised physical structure, it was not scalable to larger and robust applications. This lack of scalability has proven itself upon the realisation of the first version of protoDECK. Connecting all 168 microcontroller nodes on one data line has become a technical difficulty that ultimately has been unsolvable with the initial project budget and available expertise and the initial version of the project has been realised (by Marco Verde and MakrDavid Hosale) as w physical assembly of components and with operational individual nodes, but without the networking or control capabilities of the installation as a whole. In consequence no dynamic behaviours or interactions of protoDECK were prototyped.

The second iteration of the protoDECK project has been taken over by the author in close collaboration with dr. Stefan Dulman, Andrei Pruteanu and a group of students at the faculty of Computer Science, Embedded Software, TU Delft. The approach taken was to replace the central "bus" connection linking all tiles to one central controller, by local data connections between tiles. Consequently, each tile would communicate only to its neighbours. The original electronic hardware provided strong limitations and generated a high percentage of errors in the communication (up to 40% of messages contained errors). Nevertheless, various simple emergent light patterns were designed, implemented and tested with several users. The experiments have shown the validity of the approach and have led to the redesign of the hardware system to support fast and error-free local communication between nodes.

The latest version of the hardware allows additionally "viral programming" of the tiles, whereby uploading a behaviour to only one tile, this behaviour can spread to other indicated nodes through the distributed network. Consequently the protoDECK 2.0 platform is intended for facilitating further experimentation and development and testing of multi-component and multi-user interactions for a variety of projects.

c) LEDworks

The LEDworks has been initially designed and prototyped as part of the sCAPE project. It is a network of light sources, emitting RGB light, thus in that respect similar to the protoDECK. However, LEDworks are mobile devices embedded with microcontrollers, which can connect into three-dimensional structures. Users can detach individual nodes and re-connect them in different configurations. Providing there are enough LEDwork devices at hand, large and complex structures can be built and transformed directly by the users.

Each node is equipped with an accelerometer and a gyroscope, meaning that it can sense its orientation and rotation once picked up by a user. Based on this capability, students have designed a number of behaviours. The ultimately selected one changes the hue of the node colour as it is being turned. Consequently nodes exchange colours with each other once connected and attempt to average the hue values. If only two nodes are connected, they ultimately both reach the average hue. If more nodes are connected, more complex patterns appear, with propagating waves and loops emerging.

Throughout the design and prototyping of the interaction between LEDwork nodes, the prototype had also been tested with users. Initially these users were designers, later tutors, ultimately visitors to the Delft Science Centre and to several other venues where the project has been exhibited. This testing has allowed verification of design assumptions and has led to improvement of programmed behaviour.

The main goal for designers was to achieve a playful experience for the users, triggering social interactions among them. For these interactions the LEDwork has proven to work as a catalyst. Several patterns have been observed. Among them were the teaching patterns, where users who would observe a certain behaviour or a surprising occurrence emerging upon interaction with one or more nodes. Common reaction was a desire to share that discovery and/or to teach just arriving visitors about the discovered functionalities. Another pattern was than among adept users, where more elaborate explorations and games were played collaboratively by modifying together interconnected nodes and creating surprising light patterns.

d) GEN

In the gen project main effectors were kinetic and included ten inflatable floor cushion elements and three bending wall elements, which were additionally equipped with LED light strips. Similarly to the firs version of protoDECK, the behaviour of the GEN installation was centrally controlled, but operated in a distributed manner through a multi-agent system of virtual agents.

The goal for the interactive process was to provide the best spatial affordance for "lounging" being a mix of relaxation of social activities. The specific requirements for lounge space can vary depending on individual preferences of users, the social context of their visit, time of the day and year and others. In order to fulfil that vibrantly changing requirement towards spatial affordances, the designers proposed a system of a "kudo economy", where each installation component would continuously attempt to provide the fittest state for the users, validated by being sat on. Consequently, components would exchange information about their state with their neighbours and locally convince their neighbours to change their states to the remembered by them successful ones by virtually trading earned kudo currency for influenced state change. Consequently their neighbours would also remember the states that gained them kudos as successful and continue the process of trading with neighbours.

Although the "kudo economy" system has never been brought to realisation due to technical problems with the installation, its simulation has shown to be very promising. Compared to protoDECK being an agenda-free platform and LEDworks building up complex interactions through basic reactions between components, GEN added complex interactions between components, where through transactions a learning process takes places. As a result, the system is theoretically able to identify patterns of user behaviours and, both as a whole and as individual, spatially located components, learn to actively adapt to these patterns.

What has been observed in the "Wizard of Oz" user testing process is that user needs to sit on GEN landscape of cushions was generated proactively by the installation. Most users had no prior desire to sit down and socialise, but certain configurations and light patterns encouraged them to do so nevertheless.

e) d|e|form

Author's d|e|form project conceptually builds upon lessons learned in earlier discussed design experiments. It set out with the goal to add the co-creation to the range of interaction types employed in iA systems, while further empowering iA system capability to evolve over time.

Earlier discussed projects showed a limited ability to evolve and adapt before prescribed capacities, as the number of components and their physical qualities have been fixed. The d|e|form project was initiated with the idea, that throughout the lifespan of an iA project, users themselves can contribute to re-design, transformation, fabrication and assembly on new components. In this way, where learning artificial agents cannot self-improve anymore, human agents step in and improve or replace the artificial agents.

The core feature of the d|e|form installation was the streamlined fabrication process of individualised components. The setup has been envisioned where the virtual agents trigger generation of new components where they are most needed. Subsequently fabrication data is sent to the CNC facility and the required elements can be fabricated. However, one or more users has to approve the creation of a new components and engage herself in the assembly process, including bringing the component to its deployment location and installing it. In this way the act of fabrication becomes a vital part of user community-architecture interactions through the co-creation process.

The second part of the co-creation process is the modification of the behaviour, sensing and effecting capabilities of components. The d|e|form system has not reached the fully functional stage. However, this process has been tested in a workshop with interactive environment minor students. In the workshop student groups were given pre-made non-interactive panels and were asked to develop a socially engaging interactive feature for children embedded in the panel. As a result three different new panel types were prototyped and tested, extending the library of possible component types.

In this way, a realised project deployed in public space can be actively improved if instruments and cultural and societal means are provided for users to add to transform or add parts of the installation.

f) Evaluation

The traced projects demonstrate a gradual increase in complexity of the many:many interactions in the designed iA systems. From initially reactive systems, projects become complex ecosystems of heterogeneous agents. Once the technical problems are solved, the main challenge of realisation of interactive architecture becomes how to steer the aggregation of such diverse interactive processes into continuously growing and flourishing ecosystems.

The main problem encountered is the inability to test and verify in a virtual or laboratory setting the performance of complex interactions involving human agents and input from outside the system. These complex interactions also are difficult to be designed from scratch and have been developed through gradual aggregation of individual 1:1 interactions. Consequently, new design and development strategies are needed for development and evolutionary improvement of interactions throughout developed projects.

3.3. Evolving interactions

a) Outset

The projects investigated in previous points show the increasing complexity of involved interactions. Designed interactions ultimately point towards self-learning buildings, where users can actively participate in guiding and executing the building transformation. The process in which iA interaction design has progressed can be summarized as a leap from cyclical linear interaction scenarios, towards individualised interactions, where no two interactions are the same. The third leap is to allow for gradual evolution of the interaction processes themselves as an iA project develops. In this way a project can be initiated by designing and prototyping simple reactions to gradually increase the interactivity degree an ultimately surpass the complexity of interactions envisioned to this point.

Just like an evaluation of iA system's performance is required in order to improve that performance (either by designers or by agents in the system), evaluation of the interaction processes is equally required to improve these processes. Rudimentary questions logically follow. What are the criteria for measuring the quality of interaction? How can interactions be continuously measured and improved integrally within the designed system?

b) Process

In traced projects, the majority of feedback about interactions has been collected by designers. It included iterative collection of surveys on users, recordings of user testing and gathering of other materials that could be later evaluated. Such methods, however, could only be executed periodically, as collection and analysis of such data is time consuming. Attempts were also made to collect feedback in a more streamlined and continuous manner. For this, the porotoTAG platform has been used to allow prospective users to provide localised feedback on objects of interests to which they had a direct connection. In that case, the feedback has been scarce and rarely used; however, this can be partly attributed to the functional shortcomings of the protoTAG platform, namely lack of user-friendly interfaces to provide information. On the other hand, lack of direct incentive for users to provide feedback on projects and filtering of relevant information were other significant issues.

In that respect, co-creation models provide a more direct ability for users to provide "tangible" feedback by directly modifying aspects of the interaction processes that don't suit user's preferences.

Following the two models of improving interactions through a) feedback to designers b) feedback to users, the third model is c) incorporation of feedback on interaction quality into the building components. That possibility is at this point mostly hypothetical, but technologically feasible. Similarly to transformation of component physical parameters, the interface and behaviour can similarly be improved. This can be started by incorporation of reactive mechanisms in switching between several interactive behaviours based on sensed conditions. Based on evaluation of that strategy, such behaviours can be further multiplied and differentiated to eventually form an evolving behaviour.

c) Evaluation

The process of evolving iA interactions is yet at its starting phase and requires further research to reach practical applications. Nevertheless, the direction and aims of this research are clear. Similarly to how individual and collective behaviour of humans evolves over time, behaviour of building components ought to be continuously improved as iA systems develop. The process of forming this behaviour can progress analogically to the organisation of this section; starting with definition of 1:1 interactions among diverse kinds of system components, consequently aggregating these interactions into larger networks, to ultimately cover the entire system and its context. Such created interactions are not intended to be permanently defined, but are expected to evolve with the iA system, which evolution can be driven by self-learning agents and by user-community modifying, removing or adding new agents to the system.

In this process, initially designers are catalysts of the formation and evolution of interactions, consequently users take over this role and in the future building components are likely gain autonomy to self-improve interaction processes.

This development needs to be approached gradually. The risk at stake is that interaction processes can evolve in undesirable directions, where they can potentially perform against the well-being of their users and hinder the global development of the iA system they belong to. Self-control feedback mechanisms (possibly similar to Asimov's laws of robotics) would need to be gradually incorporated into these components to prevent such situations.

3.4. Discussion

The initial scrutiny of user:component interactions shows that a step from reactive operation to true interaction is a difficult one. Not only so due to technical difficulties, but also due to cultural conditioning of humans expecting to be able to exert control over artificial devices. Giving up this control proves to be a problem not only for users, but also for designers. The problem of control also relates to the issue of functionality. In interactive system an affordance which interferes with users immediate desires can be provided to them to pro-actively stimulate them to induce certain activity that would ultimately lead to user satisfaction. However, such action is typically associated with limitation of freedom, since someone else's control is assumed by users to be affecting the change. The risk at stake, is that the generated condition may in fact be harmful or otherwise negatively affect users (e.g. by scaring them). The challenge of design of user:component interactions lies thus in finding the appropriate balance between responsiveness and pro-activeness in designed interactions.

Component:component interactions can be initially overlooked as they can be avoided in simpler setups where the entire installation works as one agent. However, projects such as protoDECK, GEN, LEDworks or d|e|form show the potential of iA system self-learning through the build-up of knowledge through the process of interaction[1]. Similarly, the interactions among users ultimately determine the success of a project, given that architecture operates as a catalyst for social transformation. Consequently, the performance of an iA system is the result of an aggregation of a very high number of interactions among all kinds of its components. Regardless of component learning ability, in numerous cases, user modifications to components may be required at different stages of project development. The role of users can vary, from complete redesign and addition of a new component kind to an existing installation, or serving a facility role in fabricating and assembling components generated by other artificial components (both cases explored in the d|e|form project).

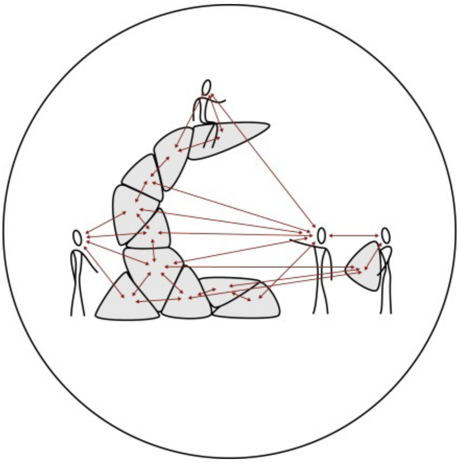


Fig.62. Ideogram: distributed interactions in an iA project

In case of autonomous self-learning components, as well as user involvement in development of iA systems, accidentally or intentionally malicious activities can occur (such as destruction of a prototype in the d|e|form installation due to miscalculation of stresses, or deliberate breaking of GEN cushion by playing children). In systems with a multitude of interactions, not only behaviour of individual components, but also emergent properties such as oscillation or any other positive feedback loop may set the system out of balance and lead to its destruction. Therefore mechanisms need to be implemented to prevent system destruction at all scales and ensure its stable development.

While 1:1 interactions are possible to be initially imagined by designers and subsequently consequently implemented, the complex ecosystems of interactions are impossible to be anticipated and need to be designed, realised and implement gradually and iteratively. No design methods exist to organise this work. Therefore new best practice examples, working examples of iA interaction patterns and corresponding techniques and instruments require further development.

Such development can only be advanced through larger case study research projects, where larger user groups can be involved and real world scenarios can be tested.

[1] Pask, *Conversation Theory, Applications in Education and Epistemology*.

Conclusions summary:

- Design of interactions involves finding a balance in direct responding to user needs and desires with pro-active provision of not directly required affordances, which ultimately lead to larger user satisfaction and/or better development of the iA system as a whole.
- Autonomously adaptation and evolution of agents can be achieved through the process of interaction, or overruled by users transforming, removing or adding components in a one-directional process, with combination of two approaches possible.
- Mechanisms are required to prevent malicious users and malicious artificial components.
- Design knowledge requires further build-up in respect to many:many interaction design patterns, techniques and instruments.
- More case study research involving larger user communities is required to investigate ways in which to advance the complexity of interactions in iA systems and to accelerate their evolution within and across projects.

4. Towards cross-project evolution

Challenges:

- Aggregation of networked building components and development of interactions between these components requires better structuring in order to deliver fully functional complex adaptive interactive architectural systems.
- Design methods, design instruments and realisation processes require further integration.
- Exchange of knowledge, techniques, system interaction patterns and strategies for guiding the development of iA systems needs to be facilitated across projects.

Previous sections have demonstrated the complexity of realising iA systems. Design concepts and virtually pre-developed iA systems provide first steps for realisation of iA systems in their destined contexts. Projects traced to this point through different views have not been following any predefined development strategy or method. Instead they attempted to explore various aspects of the development of iA by guided trial and error processes, the strategy chosen to cover a broad diversity of possible approaches.

It can be concluded that this exploratory phase is complete, and provided a vast range of projects, instruments and realisation techniques. However it has also been observed that further advancement in respect to complexity and applicability of student projects has stalled, with most recent projects hardly delivering improvement over older ones.

This state of affairs can be attributed to a number of factors.

Firstly, conceptual design, virtual prototyping and material prototyping are still relatively disconnected from each other. Further integration of those three aspects of the iA development process is essential to improve quality and robustness of attained results, and iA instruments are expected to play a critical role in this respect.

Secondly, knowledge exchange between designers is limited. This prevents consistent development of structured knowledge, techniques, system interaction patterns and strategies which can be employed to facilitate, accelerate and improve the quality of attained results.

Presented tracings have originated from different departure points. Yet, they have all converged in one theory and approach of intensification and heterogeneity, dissolving boundaries and hybridising systems. However, this approach is not yet ready to be applied in praxis. It has delivered methods and instruments. It showed scenarios of implementation. However, it does not tell where to start and how to proceed. It does not provide a pragmatic, cohesive framework to work in.

Throughout the entire fourth chapter, the notion of the “model” has been avoided. It has been originally stated that the concept of a “system”, by drawing a boundary and defining the boundaries of its components is already a reduction of reality and its great simplification. To use the world model in the context of systems would have meant a double reduction. A “system model” would thus mean a simplified representation of the reduction of reality.

This does, not however mean, that such reductions are always to be avoided. It only means that assumption of a system model from a start would mean a reduction so large, that it would have prohibited investigation of the details, which turned out to be deeply important for developed architecture.

Despite all their reductions, models are critically important for humans. Human brain cannot deal with the world in its full complexity and needs to make reductions of that complexity in order to act in the world. Similarly, in the process of development and inhabitation of architectural habitats, such reductions also have to be made.

Once it has been shown and analysed how architecture can be dissected into several, overlapping systems and only after it has been shown how the boundaries across space, time and scale of such systems are only a matter of convention, the activity of reconstructing the idea of an architectural model can be taken up again.

The framework introduced in the following chapter is an attempt to resynthesize a new kind of an architectural model, which would allow development of complex architectural systems without compromising their rich, dynamic nature and ability to adapt and flourish.

VII. Assembling the iA project framework

Summary

Chapters IV-VI have shown that design processes of iA systems share limited similarities with traditional architectural design processes. Throughout traced iA design processes, a variety of new concepts has been introduced to the design vocabulary and has transformed meanings of many traditional architectural design notions, such as those of a “model” or “project”. Chapter VII integrates the resulting new iA design ontology in the open and extensible “project framework” further called protoFRAME. protoFRAME is used as an ontological and organisational structure which integrates previously discussed design strategies, methods, techniques and design instruments and organises project content and planning.

Subsequently, three project templates utilising protoFRAME are proposed as starting kits for future projects. protoFRAME in conjunction with these templates lays grounds for further development of iA methods, techniques and instruments in an integrated manner, facilitating focused knowledge exchange and global evolution of iA as a distinct domain. Resulting design methods are expected to gradually build up a coherent, yet extensible and evolving methodology for complex adaptive interactive architecture. Auxiliary research directions are consequently defined in order to enable further advancement of iA design methods.

The chapter begins with a short discussion on the general concept of the project framework, its application to the conducted research and its relation to iA design methods and methodologies (section 1.). The discussion is continued by re-introducing and revising the notion of a “model” and subsequent synthesis of problems and opportunities encountered in earlier described design case study research experiments. Based on this synthesis, a set of features required from any iA system is scrutinised. From those features the core constituents and structure of the framework are derived (section 2). The details of the iA project framework “protoFRAME” are further specified, illustrated and discussed, in synchrony with earlier evaluated theories, methods and instruments (section 3). Eventually, an implementation strategy based on protoFRAME templates, involving an integration of the project framework with earlier developed iA instruments, is outlined and future challenges are discussed (section 4.)

1. iA project framework

Chapter VI concluded with explicit formulation of a need for better integration of virtual design processes, design and prototyping instruments, and realised, operating iA systems in iA projects. In order to provide strategies for such integration and further advance the state-of-the-art in techniques and methods for integrated iA system development, exchange of knowledge, techniques, system interaction patterns and strategies for guiding the development of iA systems need to be facilitated across projects.

Formulation of a project framework is an attempt to answer to those and other demands earlier identified throughout tracings of various design research case study projects in previous chapters. In principle, a project framework can be defined as a comprehensive definition of what generic iA projects always consist of, how they are structured and how its constituents are interrelated.

In this context, the word "project" can be used in both of its two meanings[1]; as "a specific plan or design" and "a planned undertaking" aimed at realisation of that plan and design. In other words, iA project means both the plan for an interactive architectural system and the process of development and operation of that system. This general understanding of the term, applies very well to iA projects that are processes unfolding over time and involving people. The convenience of this term lies in the semantic removal of the conceptual separation between design and implementation, which separation, as earlier discussed, dissolves in interactive architecture.

Consequently, a "project framework" is proposed to provide structure and vocabulary for describing and working with such understood "projects". The term "project framework" is not in common use in architectural or other design contexts. The closest to intended meaning of the term "framework" can be commonly encountered in software engineering. There, the concept of "software frameworks" is be defined as "reusable designs of all or part of a software system described by a set of abstract classes and the way instances of those classes collaborate."[2] Analogically, an iA project framework can be defined as a reusable structure for organisation of iA systems and can contain parts of an iA system which can be reused in multiple projects.

Although commonly architectural project frameworks are not explicitly formulated, they are implicitly used by architectural designers. Traditional architectural projects repetitively consist of design elements such as project drawings, engineering calculations, cost calculations, realisation plans, and participants, such as architects, structural engineers, contractors, clients and many, and ultimately also building components such as doors, walls or columns.

There were numerous attempts to standardize classification of architectural project components. Currently the widely accepted standard is defined in the Industry Foundation Classes[3]. The iA project framework does not attempt to replace such standards. It aims to provide a parallel and more open structure for organising and integrating iA systems, independent of any restrictive classification standards.

The iA project framework, nevertheless, needs to distance itself from implicitly established architectural frameworks. As identified in earlier chapters, development of complex adaptive interactive architecture requires new project organisation, instruments and ontological

[1] 'Project - Definition and More from the Free Merriam-Webster Dictionary', accessed August 14, 2012, http://www.merriam-webster.com/dictionary/project.

[2] Don Roberts and Ralph Johnson, 'Evolving Frameworks: A Pattern Language for Developing Object-Oriented Frameworks', in *Proceedings of the Third Conference on Pattern Languages and Programming* (Addison-Wesley, 1996).

[3] 'IFC Overview Summary', accessed August 14, 2012, http://www.buildingsmart-tech.org/specifications/ifc-overview/ifc-overview-summary.

redefinition of rudimentary components of architectural systems. That said, it does not mean that interactive architecture is not compatible with traditional architecture. It, however, requires a different "worldview" on architecture. The iA project framework's main role is thus to structure that worldview and to provide a basis for integration of design and realisation of complex adaptive interactive architectural systems and their further evolution and growth.

The following framework has been constructed iteratively, as it was being gradually assembled out of recursively occurring patterns and concepts across performed case studies. Several versions of this framework have been earlier proposed[1]. The following is the latest iteration, yet the manner of its formulation permits its further development.

2. Challenges

Acknowledgment of the need for an iA project framework concluded the background research on adaptability and interaction in architecture. The primary postulate for such framework is to integrate thus far scattered endeavours in development of architecture capable to actively perform as a complex, dynamic and adaptive system. As argued, such integration is a prerequisite to further develop methods and techniques for creation of interactive architecture.

Due to little and fragmented research in iA, only very general guidelines for designing iA systems could have been formulated initially at the outset of design case study research. These guidelines included recognition of the highly distributed and heterogeneous nature of iA systems and indication of most generic categories of agents constituting such systems (namely humans, building components and other artefacts, spaces and virtual agents). Beyond those general assumptions, no further constraints were imposed on iA systems, no technological solutions were implied and no strategies for system developments were indicated.

Following the preliminary guidelines, design research case study explorations allowed investigation of different aspects of design and prototyping processes of interactive architecture. These explorations provided a deeper understanding of the complexdistributed nature of iA and intricate processes of its formation. The design case study research allowed identification of major problems faced when attempting to design complex adaptive architectural systems in respect to design methods, design environments and system prototyping and deployment.

The lack of integration within and across projects has been the prevailing problem along with inability to communicate and execute frequent design iterations and the lack of established conventions and examples for iA system design. Thorough evaluation of these problems has led to the following conclusions about the challenges to which the integrated project framework for iA needs to respond.

2.1. Structure

Aggregation of networked building components and development of interactions between these components requires better structuring in order to deliver fully functional complex adaptive interactive architectural systems.

[1] Tomasz Jaskiewicz, '(In:)forming Interactive Architectural Systems, Case of the xMAiA Meta-model', in *ACADIA 10: LIFE In:formation, On Responsive Information and Variations in Architecture* (presented at the 30th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), New York: ACADIA, 2010).

a) Transparency

In the future outlook, the strict division between "developers" and "clients" of iA dissolves. In complex projects developers cease to be able to have a complete knowledge of all system constituents. On the other hand, clients may be required to take upon some roles of system developers, by contributing to re-design, improvement and customization of system components.

In order to deal with high degrees of system complexity and to allow bottom-up modifications by "clients" iA systems need to operate in a highly transparent manner. This means that iA systems are required to be consistently comprehensible by all developer and client groups. "Black-boxing" of sub-systems needs to be done consequently in the same way for all groups of developers and clients. It should allow conceptually developed ideas to be translated to their technical implementation and inversely, allow implemented solutions to be brought back to simple, commonly understood descriptions. Such transparency additionally promises facilitation of reuse of components and technical solutions.

High degree of transparency can, but does not necessarily involve open-source components. For security or copyright reasons, open-source may not be possible in many cases.

Transparency empowers trust and removes many privacy related concerns. Through knowledge of system's operation, users can opt-in and opt-out of participation.

b) Shared ontology

Selected case study research projects shown that in order to permit collaboration, developers needed to first arrive at the shared ontological model of the architectural system, from which only later differentiations could emerge. Such models would include identifications of system components, but also relations between them, and their overall purpose of existence. Similarly, project clients would often misunderstand designer's intentions due to assuming a different ontological model of the project than intended by its creators. This would typically occur due to cultural conventions and lack of technological insight. This has led to inability to constructively discuss iA systems and to identify their flaws and opportunities.

"Ontology" in philosophy is a study of what entities exist in the world, while in computer science it generally means a definition of taxonomy and vocabulary within a given domain. In view of this research the usage of the term bridges its philosophical and computer science meaning, as also broadly by ontology, we can understand an abstraction of reality into a system through identification of entities and relations between them. In case study research projects different participants clearly saw different systems while looking at the same actual or virtually designed reality. In this understanding of the term, they did not share the same ontology.

The common discrepancy between ontologies encountered in case study research projects lied at attribution of agency. Users would typically attribute agency to the entire system, designers to its conceptually identified virtual components, and engineers to specific technical apparatuses.

Predefining project ontology in a top-down fashion at the outset of designed proved beneficial only in a limited way. It faced resistance from designers seeing it as a constraint or hindrance to their creativity, from engineers as not corresponding to technical solutions, and was difficult to convey to persons not directly involved in the design process.

Consequently, the aim for the framework is to provide a generic ontological structure in such a way that culturally and professionally valid specific ontologies of different project participants could be integrated.

A key for defining such shared ontology lies in the understanding of attributed agency, tied to definition of identifiable and performing entities. Such ontology needs to account for a) existence of material and virtual agents alike b) autonomy of agents c) change of agents over time d) interactions between agents e) nested agents (where multiple agents can constitute one higher-order agent) and agents forming a semi-lattice structure (where one agent can be part of many higher-order agents)

c) Uniqueness of components

In systems where many elements share same properties it is convenient to treat those elements as identical. This convenience has been strengthened by the role of mass-production in modern culture, where things are produced in large series of close-to-identical objects. Similarly, in biological systems, individual cells in an organism initially do not differ. However, in those systems cells differentiate based on their individual history, local interaction to other cells and external stimuli. In this way a cell gains uniqueness through its autonomy. Still, cells of the same type may be indistinguishable from each other.

Many phenomena observed in design research case study systems involved individualisation of components similar to that in living organism, where component's individuality depended solely on its history and spatial proximity to other components (e.g. individualisation of elements in the cockpit building or temporal individualisation of light colour in the ledwork project). However, in many situations a global manner for identification of objects was required. Such was the case in elements belonging to multiple sets or moved between sets, while being required to retain relationships to elements outside of its set. Such was the case with any kind of component being fabricated, when it is moved a virtual model to a physical world, but needs to retain the relation to its virtual information and history, as well as it needs to be identified for on-site-assembly information.

Due to this, not only each element needs to be treated as a unique, autonomous agent, but it also requires a globally unique identification of its components (although such identification may not be explicitly needed by all iA systems or all components in a given system).

d) Level of detail

The common issue encountered by designers in every project has been the definition of the scale of basic project components. As broadly discussed in section II.1., components of architectural systems can be nested in each other. Any architectural project operates on numerous scales once realised and a single projects can involve as much design of details and interiors, as urban scale interactions between entire buildings.

Traditional designs can focus solely on one level of detail of involved components, as through established design and planning patterns (including societal, ownership or legal models) the traditionally designed, and fixed spatial interventions can be abstracted into known higher scale conventions, e.g. that of a residential or public building, and dealt with on smaller scale, e.g. by furniture designers or plumbers.

Such level of detail focus is impossible for iA projects. Due to the high novelty factor, designers of every iA system need to be explicitly aware of its operation on all possible level of detail scales. Moving from one level of detail to another is necessary in each iA project (as clearly shown e.g. in reNDSM project development, where design of component detail had direct effect on urban design scale through the aggregation of these components).

e) Stratification of system models

The complexity of iA systems means that it is impossible to represent those systems with a single model. In studied design research projects, any initial design concepts' simplistic representation became gradually "stratified" into multiple representations showing different aspects of the concept, eventually to be stratified further into detailed implementation-ready descriptions. Such models would be complementary to each-other and only together form a comprehensive description (that can be referred to as a super-model) of a designed system.

Similarly, deployed systems would consist of a number of supplementary and/or overlapping sub-systems corresponding to some of the complementary models (such systems could be for example a system of embedded components and a system of project inhabitants, or system of virtual design agents). Consequently, the iA framework is required to support co-existence of heterogeneous models and systems and their integration.

2.2. Process integration

Following earlier discussion, it can be stated that design methods, design instruments and realisation processes require further integration for development of iA.

a) Facilitation of (trans)formation of complex ecosystems

The main problem faced by all studied designers (given the challenge to design an adaptive building) was the lack of broader reference and design procedure. Traditional and conventional architectural design methods proved largely unsuitable for design of interactive architectural systems. In respect to this:

a) Designers found it difficult to establish the boundary of their projects. This includes spatial boundary, temporal boundary and lower and upper scale boundaries. Those boundaries were shown to largely vary, also when comparing projects following the same assignment.

b) The context of designed systems had strong influence on those systems. This means that designed architectural systems could not be reliably formed (as autarchic systems) in isolation from the actual location they were designed to be inserted into. Insertion of a system designed in isolation from the target context into that context would radically alter the way in which that system operates, but it would also influence the context.

c) All studied projects were ontologically approached as ecosystems of interrelated components. This approach proved highly demanding to work with. It was observed that designers found it difficult to simultaneously design systems of high adaptability, large number of components and heterogeneity of components. Accomplishment of any of the two qualities would typically rule out the third.

d) Little or no consistency has been observed among design methods and little correlation could be made between applied method and eventual quality of the design result. Too many factors were at play (such as personal motivation of designers, external input in the design process, mutual influences, skill differences) to provide a clear assessment of employed methods. Methods as different as 1) starting from analysis of design problem and quantification of design parameters and designing in direct response or 2) starting from generic design idea and adapting it to fit the specificities of design challenge, have been equally capable of delivering good design results. It can be concluded that not one design method exists that the framework should support.

Concluding the point, the project framework needs to provide a generic “canvas” for designers to work with, on which diverse design methods can unfold. This framework needs to allow dealing with systems that include existing components as well as newly created components. It needs to support equally systems consisting of a very high number of components, high heterogeneity of components and adaptability of components and entire systems.

b) Many-by-many design, realisation, use

The challenge consequently faced in all studied projects was not only design of systems consisting of many autonomous components, but also constructive involvement of many human participants in the process of formation of such systems.

In all design research case study projects multiple participants were involved. The skills and time of their involvement largely varied. The two main groups were developers (designers, engineers, experts) and clients (teachers, inhabitants, visitors, observers, often also referred to as validators). Both developers and clients would interact in some ways with developed projects. Developers would have in-depth knowledge of projects and interact by modifying the workings of the system. Clients would deal with projects more superficially, by experiencing the system, but having limited knowledge of its inner workings and being constrained to interactions allowed by Developers.

Generally, it has been observed that approaching any person involved in the design process as a potential designer provides an advantage. (The term "designer" in this case can refer not only to an architectural or industrial designers, but also e.g. engineers-designers, business-model-designers, clients-designers or users-designers. The difference between a designer and any other project participant lies thus purely in this person's active application of "design thinking" in the design process, that otherwise could be referred to as creative problem solving. To define "design thinking" we can follow Don Norman: "What is design thinking? It means stepping back from the immediate issue and taking a broader look. It requires systems thinking: realizing that any problem is part of larger whole, and that the solution is likely to require understanding the entire system."[1])

The design process involving "developers" can be referred to as "collaborative design". The design involving "clients" can be called "participatory design". In collaborative design encountered collaborations included a) complementary collaborations, where design team participants provided complementary expertise b) parallel collaborations, where design team participants applied same class of expertise to different parts of the project.

In case of complementary collaborations the biggest bottlenecks lied in the lack of mutual understanding in information exchange between specialists and culturally established linear dependencies between specialists prohibiting simultaneous work. In parallel collaborations the biggest problems lied in designers' inability to constructively communicate and development of conflicting design solutions.

In participatory design, the main problem has been a) communication of design-in-progress to people not directly involved in the design process, b) generating incentives for participation in the process, c) collecting feedback and other design inputs.

The general problem involving both collaborative and participatory design has been the lack of correlation between design solutions and consequently their difficult or impossible integration.

All aforementioned problems can be reduced to lack of communication conventions, which the iA framework is required to provide in a way suitable for all design process participants.

c) Interactions between components

The autonomy of components is a prerequisite for local interactions. If system conditions allow, any two components of the system should be able to communicate and through that communication interact (perform transactions, mutually change). For communication to occur, a medium is required. For virtual agents, that medium is provided by the software environment these agents operate in. For physical agents that medium is the physical space

[1] Donald A. Norman, 'Design Thinking: A Useful Myth - Core77', 2010

and connections made in that space. Human agents communicate through space using their senses. Spatial relation between two human agents determines the kinds of communications that are possible, while technology can augment spatial qualities to augment certain communication channels or permit new types of communication. Artificial agents may use other communication media, such as signals sent across cables or radio waves. Among human, artificial and virtual agents different protocols exist that facilitate communication, making transactions among agents possible.

The framework for iA cannot discriminate any kind of interaction over another. It needs to permit interactions among agents that use same interaction protocol and can identify each other, but it also needs to account for situations where identity is determined by the medium, e.g. given spatial proximity of agents, mutual visibility, or wired connection between them. In such cases interactions may happen beyond protocol and can lead to transformation of protocols or formation of new ones.

Consequently, the iA framework needs to provide an open structure for defining an utilising communication between its components across diverse media and regardless of any established protocols.

d) Continuous development and evolution of designed systems

The process of interaction involves by its definition a mutual change of interacting entities. The aim of iA systems is for changes occurring in the interaction process to lead towards improvement of system components and indirectly of the system as a whole. This means that system components are required to be changed individually through local interactions. Such changes can propagate through other interactions to eventually affect the entire system; however the locality of adaptation is a prerequisite, as it can never be certain that a local improvement can be also an improvement on a global scale.

Aside from transformation of individual components, the iA system transforms through creation or destruction of components. Components can also be added or removed from a system by shifting its boundary, meaning inclusion or previously existing components or removal of components from a system without their destruction.

These processes equally permit system development and evolution. The development of a system in this case denotes its growth and other kinds of predicted transformation of the system as a whole over time. The evolution of the system can be seen as qualitative change in system’s performance, where growth independent of system’s growth.

The aim for the framework is to provide a system structure in which development and evolution of systems is not constrained

e) Interoperability of instruments

Chapter V traced evolution of a number of instruments employed to support design process of iA. These instruments were supplementary to each other. Each of these t instruments corresponded to one or more design models.

In consequence, instruments can be considered to be a bridge between conceptual designs and actual operation of designed systems. Design and development of the system can be performed in virtual space (e.g. multi agent simulation) and in the physical space (e.g. experiential prototype). However, the boundary between the virtual and actual is not clearly defined. As argued before, agents operating in virtual space or virtually on embedded devices constitute the actual operation of the physically deployed system.

Tools and instruments in studied case study projects have been employed to facilitate and explore the transition between virtual and actual in studied projects. Platforms such as swarm toolkit allowed preliminary conceptual system simulation, as well as deployment of the system and its connection to actual users, spatial conditions, involving online interfaces, real-time sensors and actuators.

Whereas tools can be seen as simple facilitators in the design process, an instrument is a platform that actively participates in project development, having autonomy to influence designers and affect project development and evolution. In this process common tools were used to design generic instruments. These instruments in turn were engaged in design processes by being "played" by designers; "composers" of a project.

As shown in case studies, a large variety of such tools and instruments can be simultaneously employed in a project. Different projects are likely to require different tools and instruments. What's more, tools and instruments themselves evolve across projects, as traced throughout chapter VI.

Accordingly, the challenge for the iA framework is to provide conventions facilitating interoperability of tools and instruments within and across projects, without constraining the development or evolution of designed systems and in correspondence to stratification processes of models.

2.3. Knowledge exchange

Exchange of knowledge, techniques, system interaction patterns and strategies for guiding the development of iA systems needs to be facilitated across projects.

a) Early deployment

The complex nature of iA projects, including many system components, many project participants, and continuous transformation of designed systems makes it impossible to predict how the designed system will perform, develop and evolve. The high number of interacting system components (things and people) creates highly complex behaviours of the system as a whole. Such behaviour can be to some extent simulated. However, simulation, being a large reduction of reality, cannot include all factors that may have an effect on the actual system's operation. This may include complex human factors, but also physical properties or external influences on the system. For this reason, designed iA systems have been shown to require early deployment and its validation performed in parallel to design process. Such deployment includes virtual and physical prototyping and eventual full system deployment.

The iA project framework is thus required to equally support both operation of iA. Ultimately, no clear line can be drawn between those two processes. On one hand operation of iA is required at early stages of design and needs to be closely integrated with design models. On the other hand, the self-perpetuated adaptivity of iA, means that any deployed iA system is in a state of continuous re-design in respect to its development and evolution.

b) Modularity

Section 1. of chapter VI delivers an overview of various types of building components that can be customized, can dynamically transform and can be assembled into larger architectural structures. Similarly, Chapter V provides a vision of protoKIT, being a collection of reusable instruments, combinable into comprehensive instumentatia for forming and developing iA projects. Eventually, also various methods and design techniques discussed in chapter IV in respect to entire design processes, as in chapter VI in respect to development of interactions, can be treated as modules, that can be combined in different ways to structure a process of iA development. The process of iA development to a high extent involves transformation of such generic modules to fit the highly specific conditions of the project, defined by its context and constraints.

Modules are thus primarily various sorts of system components, that share some of their characteristics with other components. At the same time, modules employed in iA developments can also be located outside of the developed iA systems, as is the case with reusable instruments or design techniques.

c) Extensibility

Analogically to stratification of models, and parallel development and evolution of tools and instruments, the framework for iA can itself develop and evolve over time. Such process has already been observed throughout the case study projects. Various fragments of the framework were assumed as “design rules” at the outset of each project, employed throughout the project and validated afterwards. Eventual validation lead to formation of design rules for following projects.

Clearly, many “design rules” were highly project-specific, while others could be generalised for multiple projects. Those generalised rules can be seen as the foundation for the iA framework. However, clearly, each design rule set is bound to have its flaws and may require differentiation between projects and may further evolve over time.

In consequence any project framework proposed as a general structure for iA systems needs to be formulated in a manner allowing its further adjustment and extensibility, allowing its specialised diversification and further evolution.

d) Independence of technology

In studied case study projects, the use of technology varied across development of a single project and across different projects. Availability of technology was a strongly constraining factor for projects. Throughout the years in which projects were developed, global technological developments such as development of new software and hardware platforms, but also commercial discontinuation of previously used platforms, have had a strong influence on projects. In retrospect, technology can be seen as a critical enabler of iA systems, but reliability on specific existing technology or exclusion of a new technology can be a factor largely constraining the development and evolution of iA systems, as well as evolution of the entire framework.

Additionally, in studied iA systems, technological solutions often changed throughout project development. Technological choices would lead to radical change in entire control system architecture during early prototyping phases.

Different designers and users involved in design processes of interactive environments shown largely diverse technological awareness and proficiency, where no obvious correlation between design quality and technical knowledge could be drawn. However, numerous cases shown that technically apt designers were less capable of divergent thinking, while less technically apt designers had problem with design convergence.

The role of technology in iA systems is unquestionably important. However, specific technological solutions fluctuate within and across projects. In correlation with the often constraining role of technology on some design aspects, it can be concluded that iA framework needs to be formulated independently of any possible technological solutions. However it needs to enable and facilitate inclusion of technology and its consequences on iA’s ontology and possibilities.

2.4. Conclusion

Presented challenges for an iA framework cover a broad spectrum of problems. They do not point to one specific framework structure, and in some cases may appear contradictory or mutually exclusive. On one hand the goal for the framework is to introduce conventions, on the other it is to allow unconstrained growth of designed iA systems. O one hand it is expected to drive the realisation of designed systems, on the other it is expected to be free of technological constraints. It is required to permit reusability of solutions, while not constraining the creation of new ones.

Throughout case study projects methodologies, instruments and deployments were investigated. The framework should permit every single one of these projects to be realised and facilitated within its structure.

3. protoFRAME structure

There are 6 key notions that have shown to be critical to understanding all earlier investigated case study projects. These concepts are: system, model, prototype, agent (also referred to as: component, module, element), relation (includes interaction, semantic relations) and instrument (in a reductive manner referred to as tool). However, it has been observed that, despite these notions' importance, among studied designers there has been little a priori consensus about the exact meaning of these concepts, nor a specific way in which they were to be used in design development. Such lack of shared definitions for these key terms and as a result, also no shared design worldview, significantly contributed to inability to deliver projects beyond a certain degree of simultaneous complexity in kind, number and transformation of systems over time. Secondly, provision of design instruments proved helpful in respect to developing a shared worldview among designers and engineers, yet various design instruments turned out incompatible with each other, creating worldview conflicts. Additionally, designers working within the ontology of a specific instrument had problems translating their projects into common language and reflecting on the "larger picture". Thirdly, developed prototypes have been highly useful to advance projects, but remained isolated from each other, and provided little increase of quality across projects.

In this section an attempt is to be made to synthesize a project framework based on requirements synthesized in the previous section. In line with the naming convention of protoKIT introduced in chapter VI, the proposed framework has been named "protoFRAME". Primarily the framework must specify the applicable definitions of 6 main iA concepts as listed above and the structure that binds them in critical consideration for simultaneous satisfaction of all 13 requirements.

3.1. Defining protoFRAME constituents

There are six notions that are fundamental to understanding and using protoFRAME. These notions have been discussed in detail in previous chapters and paragraphs. However, assembly of protoFRAME and its further practical application requires their compact and unambiguous re-definition specifically for the context of protoFRAME.

a) iA system

A system is a set of entities and relations between them.

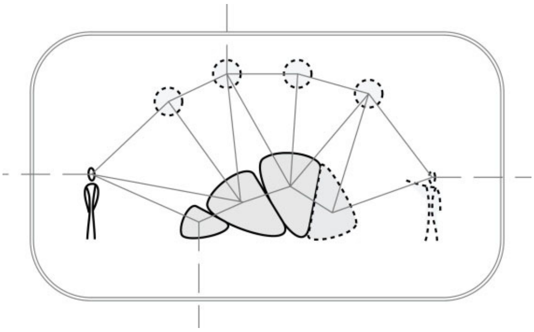


Fig.63. System as agents and relations between them, defined by a boundary and maintaining relations with its external environment.

1. An iA system consists of a large number of related to each other agents of various kinds.
2. iA systems are always open (they exchange matter and energy through their boundary).
3. every iA system has a boundary which can shift over time by including or excluding agents.
4. iA systems continuously develop and evolve over time in an agile, iterative development process.
5. iA systems include prototypes and realised projects
6. iA system is a subset of reality, in respect to the level of detail of its compoents and its distinction from the outside world.

b) iA agent

An agent is an entity in a system that is capable of autonomous action.

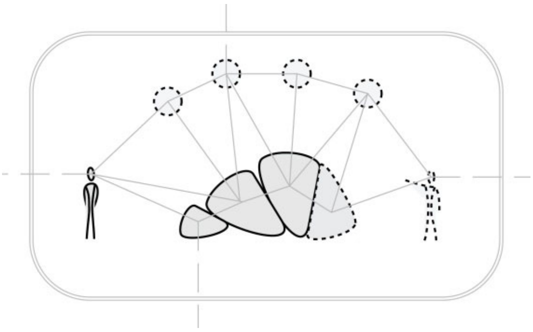


Fig.64. Heterogeneous system agents

1. Every entity in an architectural system (human, artefact, space, virtual entity) can be considered as an autonomous agent.
2. Every agent in an iA system can be adaptive.
3. Every agent is unique, but agents can have shared traits.
4. Every agent can be affected and can affect other agents in and beyond the system boundary.
5. Agents can be created and destroyed throughout system operation.
6. Agents can be added or removed to the system throughout its operation.
7. Agents can contain other agents (be systems) in a nested or in a semi-lattice structure.
8. Agents can have relations between each other.

c) iA relation

A relation occurs when a property or action of an entity affects property or action of another entity

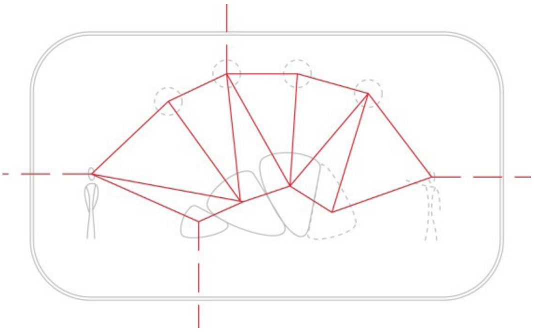


Fig.65. Network of relations in a system.

1. An iA relation can be a semantic relation between iA agents, an interaction, or dependency.
2. A relationship implies a transaction, interaction or other active dependency between agents.
3. iA relations change over time.
4. iA relations can contain other relations and agents in a nested or semi-lattice structure[1].

d) iA system model

An iA system model is a representation of selected aspects of a system.

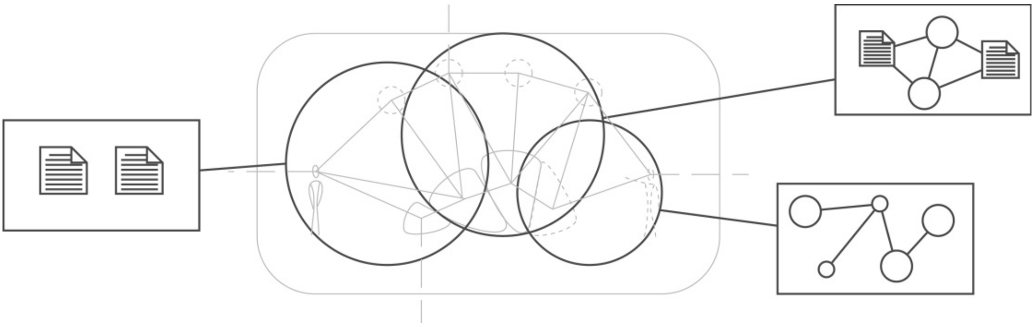


Fig.66. System models representing various aspects of the system.

1. A distinction is required between "models" that operate as sets of virtual (e.g. BIM) or physical (e.g. scale model) within the system and models that represent the iA system without being part of it. (The first group of "models" are more abstract parts of the iA system that gradually become informed using external data. The second group of iA system models are abstractions of "models" from the first group and other parts of iA systems.)
2. An" iA system model" describes iA system components and relations between them by providing an abstracted and reduced "view" on that system.
3. An iA system can be described (represented) using a variety of iA system models,

[1] A clarifying example is e.g. a relation between two agents established and maintained by these agents talking to each other using voice. The same relation can be seen as an interaction of both agents with the air medium that passes on sound.

delivering complementary views on that system

4. System models can be created before or after the system they describe.
5. System models organise systems for easier comprehension, identification of patterns, comparison, reusability, further development[1]
6. One sytem model can describe more systems.
7. iA system models can describe a different levels of detail of the iA system.

e) iA prototype

A prototype is an implementation of selected aspects of a system.

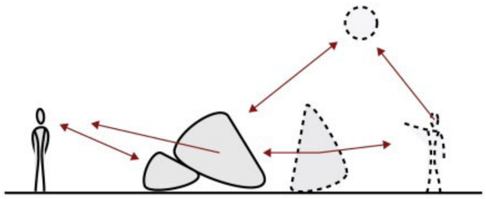


Fig.67. A prototype is the actual phenomenon that a system is an abstraction of.

1. An iA prototype is part of the actual iA system.
2. An iA prototype can be represented by one or more iA models.
3. The iA prototype provides feedback to representing it iA models.
4. Simulation is a form of an iA prototype.
5. An iA prototype is part of an iA system.

f) iA instrument

An instrument is an entity that acts as an enabler or catalyst for a certain set of processes in a system.

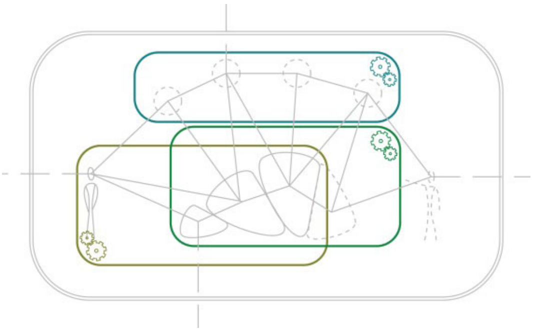


Fig.68. Instruments permit and actively influence operation of various aspects of a system.

1. An iA instrument enables development of an iA model or/and development of an iA prototype or/and connection between iA models or/and iA prototypes

The given rules serve as the foundation of protoFRAME structure. The next step in formation of a framework is further integration of these concepts into a coherent and productive worldview on iA

[1] An iA system model can be considered to be a specific kind of an agent in an iA system, since it does have an effect on that system, however the agency and autonomous performance of an iA model is limited and thus can be neglected for the purpose of the framework and clarity.

3.2. Relating protoFRAME constituents

The further specification of protoFRAME structure needs to be accompanied by deeper investigation into direct relations between previously listed constituents of protoFRAME and further clarification of fundamental dependencies and differences between them.

a) iA system - iA system model

System and system model are the two most fundamental notions present in protoFRAME. Principally, an iA system model is a representation of an iA system. In other words, an iA system model presents a certain view on an iA system. Among others, it can be a view taken in an instance of time, a view across time, a view on the organisational principles of an iA system, or some of its small detail. As established in chapter IV, architectural designers typically work with models (such as scale models, plan, section and elevation drawings, perspective drawings) that represent buildings in a singular goal-state and find it difficult to work with models of any different kind.

Theoretically, one system model could comprehensively represent an entire iA system. We can call such hypothetical model a super system model. However, such super system model would be highly impractical, as the role of models is to provide a reduced view of the system, allowing a focused insight into a specific aspect of the system. In that case even a model providing an overview of the system, is a significant reduction, as it leaves out details, that some other system models might solely focus on. In respect to this, we can say that a super system model can also be a collection of sub-models that together provide a comprehensive representation of an iA system.

A simplified distinction between a system and a model would be to state that a system is always actual and a model is always virtual. However, such distinction might be flawed in respect of depriving the model of its agency; its ability to influence the actual.

An iA model can be created before, during or after the system it represents. In all cases, the purpose of an iA model is to either create or to change the described system. Consequently, if a model has the ability to lead to change in the system, it also exhibits agency. This means it can be engaged in a reciprocal interaction with the system it represents (while not being part of that system, i.e. not being within that system’s boundary).

A significant distinction between models and systems is that systems are capable of emergence, where bottom-up behaviours lead to the performance of a system as a whole. On the other hand models, through their reductionist nature are top-down defining holistic qualities and detailed aspects of a system. These two phenomena are complementary. Top-down set direction can be used to steer an emergent system to reach preferred outcome.

b) iA system - iA project reality

iA systems are finite subsets of infinitely complex reality. In that, iA systems are “real”, but they also are an abstraction of reality. Accordingly, several systems can describe the same reality in different ways, by for example, using a different level-of-detail for defining their components.

Any system is thus a reduction of reality and this reduction is assumed by defining a boundary. A boundary defines what components belong to the system. Such boundary does not have to be related to a spatial boundary in the physical world, it can be a a boundary binding spatially separated components . An aspect of a boundary is also the level-of-detail of component definition, i.e. meaning that sub components or super-components of certain detail are no longer included in the system. Consequently, entities that are part of a component being within a system boundary can be outside of that boundary[1].

An iA system is thus a reduction of infinitely complex reality into limited and definable components and relations between them.

c) iA model - iA prototype

As shown in chapter VI, prototypes play a fundamental role in development of iA systems. However, confusion between the concepts of a model and prototype has been commonly observed among designers. The confusion originates from the broad use of architectural scale-models. A model is a representation of a system, which in architecture typically precedes creation of an actual building. A prototype, on the other hand is an early and often limited version of the actual system, deployed in order to provide feedback to system creators. The fundamental difference between the two concepts is thus the purpose. If the purpose is to represent an iA system, principally in order to communicate the system between humans, then the means of communicating that representation is a model. On the other hand, if the purpose is to create a working system, it simplification (also in terms of reduced scale) or its part, we speak of a prototype.

Following this logic, there is factually little difference between a prototype, a simulation and the actual system. An iA simulation can be considered to be a specific kind of an iA prototype. Consequently, an iA prototype is an iA system in its preliminary and simplified stage.

d) iA model - iA model

The main purpose of a model is to communicate and analyse the system, either to improve the existing system or to develop or modify a different system. For this reason, we can consider relations between models describing more than one system. protoFRAME defines three main relationships between iA models, namely correspondence, complementation and alternativity.

Two iA models correspond to each other if they use same or similar conventions (i.e. two plan drawings in the same scale, two diagrams using same symbols to describe similar phenomena) and can thus be easily compared. Two models can correspond to each other regardless whether they describe the same or different systems.

One system can be represented by multiple corresponding and non-corresponding models that show e.g. different states or aspects of that system. In such case, models are complementary to each other. This means that adding such models together builds up a more complete view on a system.

On the other hand, if two models represent different systems or different variants of a system (thus effectively also two different systems), these models are alternative to each other. For additional clarification, models that describe two different states (i.e. alternative states) of one system are complementary, not alternative to each other, while two models that show same configuration in two different systems (e.g. constituted of different components) are alternative, not complementary. Both corresponding and non-corresponding models can be alternative to each other, however the alternative character of models may be significantly more difficult, if not impossible, to determine if these models do not correspond to each other.

[1] E.g. a system of houses does not include bricks that these houses are made of.

e) iA model - iA metamodel

Two iA models can only be compared if they describe same iA system aspects (component kinds and relation kinds) and if those descriptions share the same or similar convention. In such case, these models correspond to each other. Conventions shared across models can be defined in a metamodel. A metamodel is a specific kind of a model that describes what models in a given family of models consist of.

The concept of a metamodel is very useful in the context of designing iA. In studied project cases it was shown that lack of exchange of knowledge between projects posed a significant problem for designers. This is partly because models such as perspective drawings or plans for which conventions are traditionally established were not sufficient to exchange information about iA project development and behaviour processes. On the other hand, scripts which are models of computer procedures could be easily exchanged providing that designers used same design environments and similar scripting conventions. Here, the syntax of a scripting language and its semantic structures constituted a metamodel, while e.g. a naming convention for parameters agreed upon by the group of designers extended that meta[s]model.

f) iA model - iA instrument

Use of an instrument in a design process, such as different instruments of protoKIT, typically involves creation of a specific kind of a model. Therefore design instruments inherently involve a meta-model, being in that case a convention to which all models created in an instrument comply. These conventions vary. A perspective drawing can be created using many techniques, and can offer different degree of detail. On the other hand, a computer 3d model offers a very high degree of both detail and specificity (although exceptions can be made for e.g. 3d sketches).

It has been observed that designers often resist accepting meta[s]models they have little or no experience working with, preferring to work with already known to them metamodels. Using a new metamodel follows a learning curve and leads to confusion and misinterpretation of metamodel conventions. Use of instruments accelerates that learning and allows to minimize the possibilities of misinterpretation, by constraining the models to always comply with set conventions.

An instrument can be seen as a means to facilitate, accelerate and conventionalize development of models of a given kind and increase their complementation. On the other hand, use of a certain instrument has a reciprocal effect on the developed system.

g) iA system - iA instrument

The employment of an instrument, may also directly lead to an intervention into a system[1]. This can happen directly or in combination with creation of a model. In either case, an instrument provides means for specific kinds of transformations of a system.

A distinction can be made between instruments for making models, let us call them “modellers”, and instruments for applying changes to working systems, let us call them “deployers”. However, this distinction can disappear, when the two types of instruments are merged into a singular instrumentarium, which can occur to the point when the difference between the two is indistinguishable. Such is the case with modern cad systems, where component models can be directly translated into production files and sent to production machines in a seamless process.

[1] Even in most traditional example, a piano is an instrument, playing on which, if successful, changes the system of listeners by changing the listeners themselves by providing a deep experience, possibly establishing a special relationship between the player and listeners. Score can be used as a model of music to be played.

h) iA system meta model - iA framework

The last conept that requires a closer revision is this of the iA project framework itself. In presented view, such framework is a special kind of a metamodel, which defines most generic conventions for the use of the concept of system, instrument, model, metamodel, agent and relationship and other discussed concepts. However, as much as other metamodels can change across projects or even throughout one project, the framework as a meta-metamodel is expected to only gradually evolve across projects.

3.3. Organising protoFRAME constituents using degrees of abstraction

Organising protoFRAME cocnepts into degrees of abstraction aims to provide a clear overview and distinction of different aspects of the framework, and to facilitate its practical application. A layer is here defined as a container for concepts that share a similar degree of abstraction. This structure is inspired by Model Driven Architecture (MDA) software design approach developed by the Object Modelling Group (OMG). However, unlike MDA, protoFRAME departs from a significantly higher level of abstraction in initial model description, has a flexible model structure, does not separate virtual from material (software from hardware), and does not include detailed standards for representation or other modelling formalisms. It also takes inspiration in multi-agent modelling techniques. However, its structure is more general than existing multi-agent frameworks. It is also purposefully detached from hardware and software technological constraints and conventions.

The two fundamental degrees of abstraction that can be clearly outlined is one of a system and models. We can refer to them as 1$^{\text{st DOA and 2nd DOA.}}$

a) No abstraction – iA prototype and realised building

The reality is infinitely complex. This means that real phenomena have no defined level of detail nor spatial or temporal boundary. We can continuously and infinitely zoom in and out, traverse space and time. As humans, in order to comprehend and function in reality we need to understand and operate in- and on it through defining temporal and spatial boundaries of its parts and thus developing finite and comprehensible abstractions of reality. In other words, an abstraction of reality is its simplification, that allows us to deal with its infinite complexity.

b) 1st degree of abstraction - iA system

The system layer of protoFRAME contains the actual iA system, which is open by being able to exchange matter and energy with its context, consists of iA agents and relations between them, limited by a boundary (as further defined in 2.1.a.). A system can include prototypes (deinfed in 2.1.d), which also can be simulations. Its agents (defined in 2.1.b) can be of various kinds (including immaterial agents). An iA system can be shaped by instruments (2.1.f).

The fundamental building block of a system is an agent. Earlier defined iA agents include humans, artefacts (e.g. building components, devices), spaces, virtual entities. This classification is very general, and further classifications can be made, into e.g. mobile or fixed agents. Yet, such classifications can be counterproductive, as they lead to constraining of development of new agent types, by making it easier to use agents that fall into well-defined category.

Instruments and models can be interpreted as agents if a reciprocal relation is established between other parts of the system. Such can be the case if users of the systems use instruments and/or models to transform the system. In such cases, models and instruments can also exist as parts of the actual iA system. However, if a model or instrument does not constitute an inherent part of system's operation, it should not be placed within system's boundary.

Inclusion of human designers and experts as agents in developed systems involves a similar dichotomy. By default designers and experts are not agents in designed by them systems. However, through, for example, experiential prototyping, they can put themselves into roles of system users and directly interact with system components, thus becoming themselves agents in the system. On the other hand, in participatory iA systems, all users may play an active a role of designers or experts.

In conclusion, typical iA systems consist mainly of interacting building components and devices, to a lesser extent users, spaces and virtual entities. All iA systems are open systems, meaning they exchange matter, energy and information with their environments. Design instruments, models, designers and experts are to be included in systems only if the planned operation of the system actively involves their participation. Otherwise they can be treated as input from outside the system.

iA system seen as the 1st degree of abstraction of reality protoFRAME is to be used when focusing on the actual operation of the system, i.e. when working, designing, experiencing, testing the iA system during its operation, thinking "from within" the system, talking about actual scenarios of system operation. The specificity of layer-1 is that it is bound to the present moment in time, when defining the actual state of the system. In this, the past and future of the system lies in individual agents and relations between them. As the systems grow complex with a high number of and increasing heterogeneity of agents, without the use of models and abstractions, an overview and long term planning of the system is difficult to achieve from within the system.

c) 2nd degree of abstraction - iA system models

The second degree of abstraction of protoFRAME contains all models directly describing the given iA system. The purpose of these models is to provide a comprehensive description of the system. Such description can relate not only to the present state of the system, but also to its past or future, as a result also allowing description of system's development and evolution over time.

As discussed to this point, iA agents and relations between them exist in reality as even an imagined project is real by virtually existing in the head of its designer. At the same time iA agents also constitute an iA system by being its physical or virtual components. In that the system is a reduction of infinitely complex reality, as it deals with a limited number of levels of detail of the reality (the atoms out of which a building component is made are out of concern; not in the dealt level of detail). Eventually, iA agents and relations also exist within iA system models, that at a higher degree of abstraction describe selectively various aspects of the system, again limiting the dealt with level of detail.

Consequently, two main roles of iA models can are a) abstraction of systems components and relations for easier comprehension, identification of patterns and comparison of systems b) study and design of system's operation over time. Neither of those two tasks can be performed from within the system in an instance of time and without the use of models. For these tasks a view from "the outside" of the system is required.

Abstraction of systems through models means that a model provides a reduced (simplified) view on some aspect of a system. Such view can be easily compared to alike views of other systems. It can also be modified, without the need of going into all detail of a system.

Study and design of systems over time means that models can be used not only to describe structure and relations of a system, but also to describe behaviours, development and evolution of iA systems. Through such models behaviours, development and evolution of iA systems can be analysed and compared. Consequently behaviours, development and evolution of iA systems can be also designed, which task requires modelling of future systems, and the processes of their creation.

Traditionally architectural design uses models for representation of static systems, almost entirely neglecting behaviour, development and evolution over time. When comparing to models used in UML, such models would belong to the structure diagram category, leaving behaviour diagrams, and interaction diagrams without equivalent architectural models.

Analysed projects show clearly that entirely new kinds of models need to be developed in order to successfully design iA systems. Lack of such models makes advancement of iA projects difficult, due to lack of common abstraction that would allow comparison of iA systems, and consequently perpetuate their evolution.

d) 3rd degree of abstraction - iA system metamodels

1st and 2nd degrees of abstraction relate to two fundamental activities in the iA design process; system operation and its design through modelling (including analysis and design). However, in order to perpetuate development of iA, conventions for models should be shared between multiple iA systems and iA projects. Meta-models provide thus a 3rd degree of abstraction. While models are expected to be continuously modified throughout the development of a project, metamodels are not expected to change, or change only slightly. A metamodel describes what a whole family of models consists of. Therefore, for any model to belong to a family of models, that models has to conform with the metamodel of that family.

In case study projects certain models have been recurring, leading to a gradual emergence of iA-specific model families. These model families can be grouped in two categories, namely structure models and behaviour models. Structure models describe how an iA system is organise, what it consists of, what are its boundaries. iA systems have a very large number of states, therefore models exist that relate these states to each other. Additionally context models are necessary to establish relationship between iA system and its context, thus entities that are not within system’s boundary.

o State models (Structure Models)

- All state models provide distinct “views” on specific states of an iA system at a given moment or over a certain period of time. They can exist in parallel and describe different aspects of the iA system state, or they can describe different states. Some state models can change more often than others. Changes in one model can require changes in other parallel models describing the same state and may have an influence on other states. Models are linked, through the system they describe, including interconnecting all levels of detail at all times.
- context model (the definition of all entities having potential influence on the iA project, but not being parts of it)
- concept model (a clear, and quantifiable vision defining the development direction and goal of the iA system)
- mass model (the model of the architectural form informed by the concept, context and other models)
- spatial model (the model defining the spatial relations within the project, comprising of both open closed spaces and including mobile spatial preferences that users take along with them)
- network model (the model of non-spatial relationships incl. grouping, semantic, dependency, open to changing data but with a designed structure)
- building component model (model of an individual component)

- building component assembly model (model of an aggregation of components)
- social model (including actuators and users, all with unique identities
- inventory model (inventory of all agents in the iA system)

o Transformation models (Behaviour models)

The second group of iA models are temporal transition models, that describe how system changes over time, inlcuding such aspects of the iA system as e.g. adaptation, growth and dynamic transformations. This category includes models of behaviour of individual system components, models describing the flow of energy and matter through designed systems, local interaction scenario models (use-case models), as well as general models for system development and its predicted evolution.

- State transition model
- Component behaviour model
- Internal/external matter and energy flow and transaction model
- State use-case model
- Development model
- Evolution model

e) 4th degree of abstraction - iA system meta-metamodel

For a constrained problem domain, a project framework could have been limited to a defined set of metamodels, fixed models and implementation instruments. However, in case of iA systems, design research case studies have shown that such approach would be too constraining. In investigated projects, no single design strategy dominated. iA as a young domain needs to mature, and in the process numerous approaches, accompanied by different metamodel sets need to be investigated, compared and eventually evolve into new sets bringing higher quality systems to life.

For this reason, protoFRAME provides a meta-metamodel, which is the definition of concepts and relations between them presented in this chapter. In that it is self-recursive, as it describes itself as layer-4 in its own structure.

f) iA entities across abstractions

Various degrees of abstraction allow description and design of an iA system at various levels of detail, both in respect to scale and time. The constituents of iA projects, namely iA agents and iA relations “exist” at all degrees of abstractions. In other words, iA agents and relations have infinite amount of detail in reality. They have all detail required for their design and operation at the 1st degree of abstraction, as parts of an iA system. They are described with less detail and selectively in various models at the 2nd degree of abstraction, exist as groups of specialised entities defined in metamodels. Their nature is most generically described at the 4th degree of abstraction in the iA metamodel, which serves as a definition of the ontological “worldview” for all iA projects. As discussed in more detail in point 3.5, iA instruments can also exist and operate at several degrees of abstraction, which can (but don’t have to) be shared with protoFRAME.

g) Overview

In summary, protoFRAME organises notions constituting any iA development process (from design to use) by distinguishing four degrees of abstraction (DOA), to which these notions belong. The real world in iA projects are set do operate and develop is infinitely complex, and has no limit to the number of parts, their kinds or scope of view in respect to size or timescale. An iA system defines countable and identifiable discrete parts out of reality, in that providing a comprehensible boundary in which users and designers operate. Any complex iA system has thus a limit to its scope and level of detail. It is therefore defined as belonging to the 1st DOA. A system, in turn, is described using multiple "views" (models), where each view focuses on a subset of the iA system (in respect to scale, time, scope). System iA models constitute the 2nd DOA of the iA project. Shared conventions for creating iA models that allow sustain existence of these models and improvement of their efficiency are parts of the 3rd DOA of protoFRAME. The 4th DOA defines the "worldview" that all iA projects are developed with.

All DOAs of an iA project can be changed over time within one project, or across many projects. However, the worldview of iA (4th DOA) is expected to remain relatively constant, while metamodels (3rd DOA) are expected to change slowly, based on feedback from developed iA projects. The iA models (2nd DOA) and iA system (1st DOA) are where the main attention of designers and users is focused on. The iA system changes continuously and rapidly. Yet, iA systems develop too complex to be easily comprehended in their entirety. Multiple iA models describing an iA system provide designers and users with means to more tangibly understand and alter the dynamically changing iA project through selective "views" on limited scale, time or scope of the complex iA system of that project.

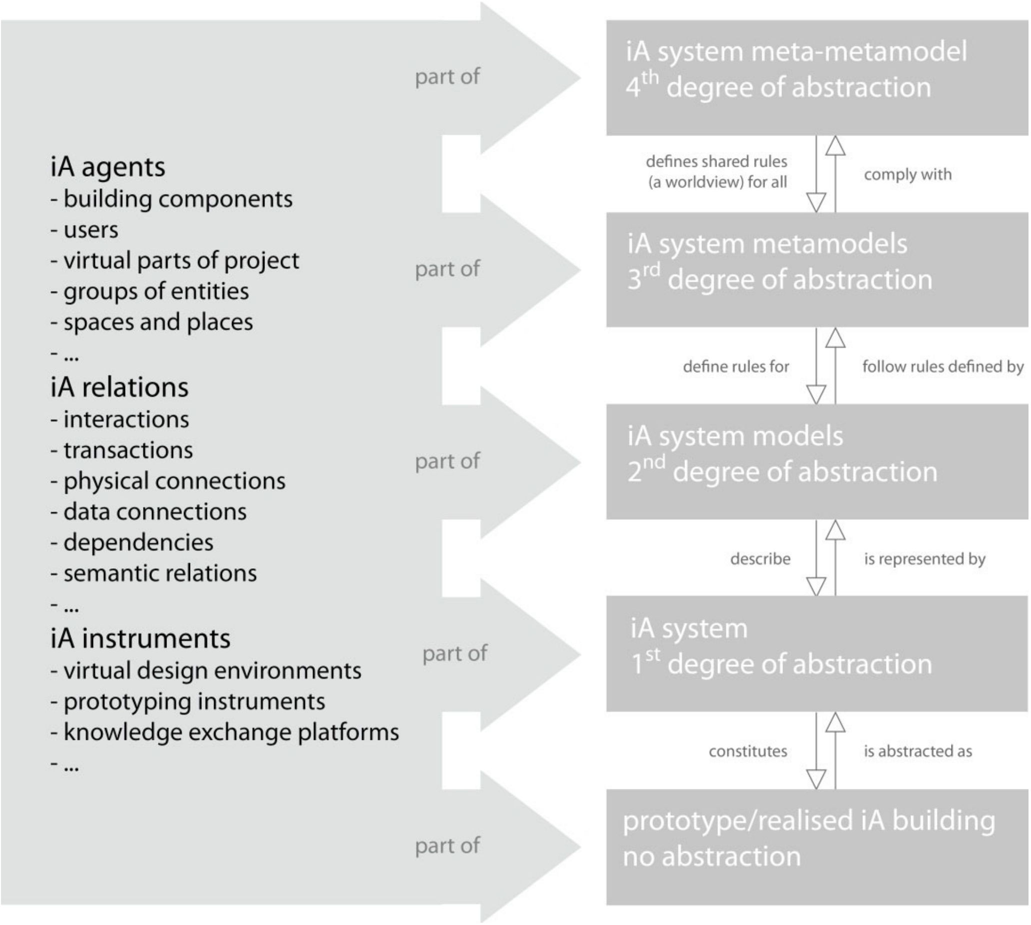


Fig.69. Integration of key concepts of protoFRAME using degrees of abstraction

3.4. Stratification of systems and models into layers

Typically, iA design processes start with one simple model, usually a sketch or a diagram. Conventionally, this model provides an initial general outline of how a design brief is to be answered and is followed by more detailed models. In practice, the initial model in a design process can describe any aspect of the designed system. The starting point of a project may thus be a general strategy towards development of the designed architectural system, but it can also be its detail or a specific technique or technology to which other aspects of the system will be subdued. Thus, no conventions can be assumed as to the type of the initial model.

As the system develops, the one initial model differentiates into multiple complementary models depicting the entire system. In this process the initial model may be removed from the project, replaced or transformed.

Observed development of iA models, especially in the early phase of the design process, is rarely linear. Design processes are often restarted, steps back can be taken, and numerous alternatives may be developed next to each other. The rationale behind system development may be fuzzy, often influenced by factors from outside the system domain. Designer's intuition may play a significant role in the decision making. As a result, models can differentiate in unpredictable ways, while new types of models are being introduced or removed in the process.

Such process of differentiation of models can be referred to as "stratification". Coexisting families of models can consequently be grouped as belonging to separate "layers" that can also be called "strata". Layers are continuously transformed, merging, splitting, appearing and disappearing. Hence, the term "stratification" is used to signify a fluid process of layer formation, where layers are flexible and fuzzily defined, unlike degrees of abstraction, which provide a rigid, clearly differentiated classification for project constituents. Within one degree of abstraction, multiple layers are formed, each layer containing a model describing a different sub-system of the developed iA system.

Stratification of models is reflected in differentiation of metamodels and subdivision of the iA system into subsystems. Whenever a new type of model is introduced an explicit or explicit metamodel is added to the project. Each 2nd DOA (model) layer corresponds to one 3rd DOA (metamodel) layer, while one 3rd DOA layer can correspond to multiple 2nd DOA. protoFRAME 2nd DOA can contain layers corresponding to any of the models listed in section 1.5, as long as these models are complementary to each other, not contradictory. Reassuming, models on different layers need to describe the same reality in a complimentary manner.

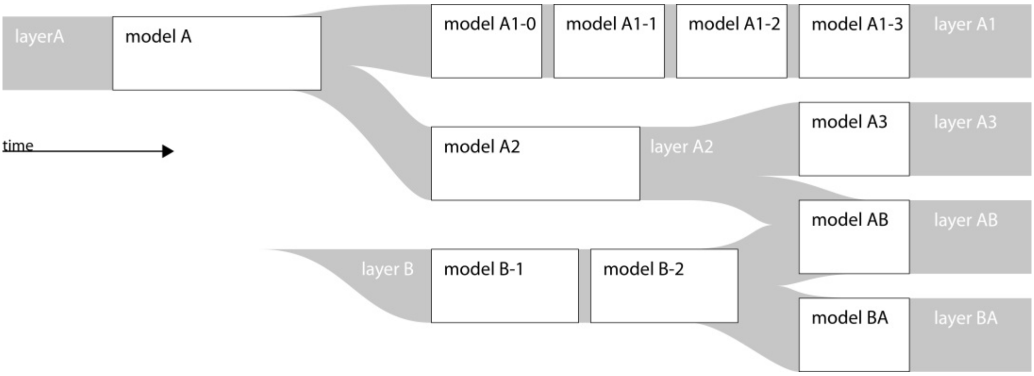


Fig.70. stratification of models in protoFRAME over time

3.5. Integration of instruments in protoFRAME

Stratification during project development is a process observed to pose difficulties to designers. It reflects the highly unpredictable nature of design processes, and inability to confine them in rigid methodologies. Shared conventions for models (hence, shared metamodels) allow to facilitate that process through reuse of design patterns, comparison and resulting improvement of models. Historically and contemporarily, instruments facilitate or permit development of design models, enforcing established conventions or introducing new ones. Only repeated use of an instrument leads to its mastery. At the same time instruments serve an integrative role in architectural design processes[1].

Furthermore, studied case projects showed a new requirement in iA design when compared to traditional architecture. That trend is the need for agile iterations in the design process, where models need to be rapidly and frequently transformed into simulations and prototypes. Some of the design instruments investigated in chapter V served those purposes. protoKIT has been presented as an extensible set of evolving instruments for developing architecture.

Consequently, three roles can be given to instruments in protoFRAME. Namely, an instrument can be used to:

- Construct models according to a certain metamodel
- Deploy models as simulations, prototypes or final realisations
- Communicate between models

These three tasks can be combined. As an example, protoSWARM is a specialised instrument to form a model of a system of autonomous components and simulate the behaviour of that model. It provides methods for importing and exporting data, which allows connecting to other models. On the other hand, protoNODE is an instrument strictly used for deploying prototypes. It imports a behaviour model (script), and subsequently becomes part of an actual system. It is not directly used to formulate a model. More conventional instruments, such as autocad or a UML charting software allow construction of models only, providing no or very limited possibilities for exchange with other models or deployment of models.

The role of instruments is critical to implementation of protoFRAME. It has been shown that designers resist imposed on them ontological classifications or methods if they don't provide direct benefits to the design process. Instruments allow significant acceleration of the design process, while providing a methodological constraints and imposing conventions and design rules that are in synchrony with protoFRAME and permit advancement of iA methodologies through grouping of model types, allowing comparison of systems and refinement of metamodels.

3.6. Design process organisation using protoFRAME

Several fundamental problems have been identified in respect to organisation of iA design processes.

- Designer in the system versus designer outside of the system; designers operate "in" the system when prototyping, working with simulations, while the traditional role for designers is top-down, without being involved in the operation of the actual system. In iA both positions need to be combined.

[1] Brankop Kolarevic, 'Towards Integrative Design', *International Journal of Architectural Computing* 7, no. 3 (September 1, 2009): 335–344.

- Multiple collaborating designers; large teams of designers, engineers, other experts and even designing users participate in iA development from its early stages.
- Engineers becoming designers; discipline boundaries blur. Engineers are expected to be capable of design thinking, and “out of the box” problem solving.
- Users becoming designers; users, including testers, clients or inhabitants, participate in the design process and actively influence the iA system’s development and evolution throughout its lifetime
- Fragmentation of the process and project; due to their complexity and multiple participants, iA design processes tend to become fragmented resulting in difficulties in integration of project’s strata.

protoFRAME sees designers as principally external to the iA system, designing the system from outside the system boundary and interacting with the project on all degrees of abstraction. This means that a designer can directly influence the system, model of that system, redefine the rules of that model in a metamodel or even change the meta-metamodel, meaning changing the fundamental worldview which all models and the iA system follow. Exception occurs when designer becomes a user, in that case becoming also part of the project and being modelled. All instruments require interaction with designers. Additional models can be created to guide project organisation. Project participants’ roles are flexible, yet are organised based on the layer on which a given group of participants performs most of its tasks. In such project organisation, project coordinators work on high degree of abstraction problems, designers directly work with the system or indirectly through its models while end users operate within the system, not being concerned with any protoFRAME levels of abstraction (creating their own, personal models of abstractions).

3.7. Conclusion

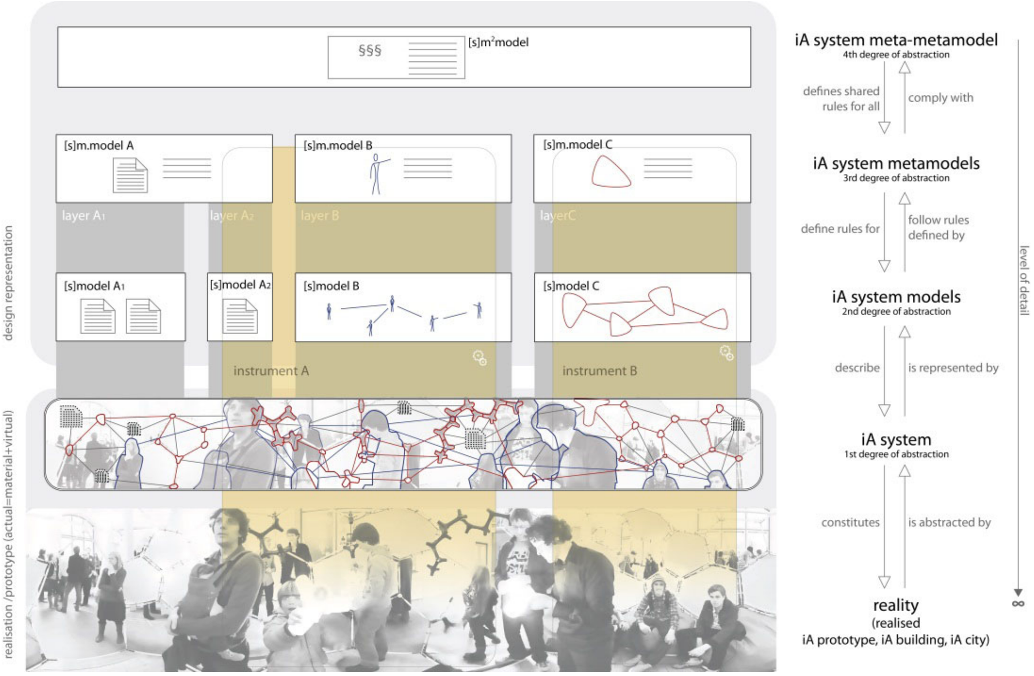


Fig.71. Layers of protoFRAME

protoFRAME provides a comprehensive structure for an iA project, which organizes its constituents across 5 degrees of abstraction, unlimited number of dynamic functional layers, inclusion of instruments in three roles and explicit relation to the designers in the process.

However, protoFRAME remains a theoretical construct without direct applications. These applications can be achieved through using templates for design process initiation. Following section proposes three templates for most common classes of iA projects. Eventually, from every realized project a new template can be extracted to initiate a new project, analogically to what has been observed in case study projects.

Any iA project can be viewed and developed through the organisational structure provided by protoFRAME. However, the structure in itself is not sufficient to develop better iA systems, in itself it is relatively remote from the practical side of iA development. It is also time consuming to break-down every iA system into protoFRAME structure if it is not develop in that structure from the start. Designers would be unlikely to understand and use protoFRAME if it would require additional effort from them. protoFRAME needs to be implemented in a way that is driectly beneficial and straight-forward to iA developers at any stage of the development process.

protoFRAME templates are introduced as a solution to problems mentioned above. A protoFRAME template is a generic, default architectural system, preliminarily pre-developed across all protoFRAME layers. Such template iA system can be used as a starting point for a variety of projects. By being generic and neutral, it excludes a design concept, thus is not “informed” by project context, designer’s agenda or specific needs of its users. The “infusion” of protoFRAME template by a design concept can thus be marked as the starting point of an iA project. The project concept should not be influenced by the template it infuses. However, the worldview used in formulating the concept should not contradict the “worldview” (meta-metamodel) of the template.

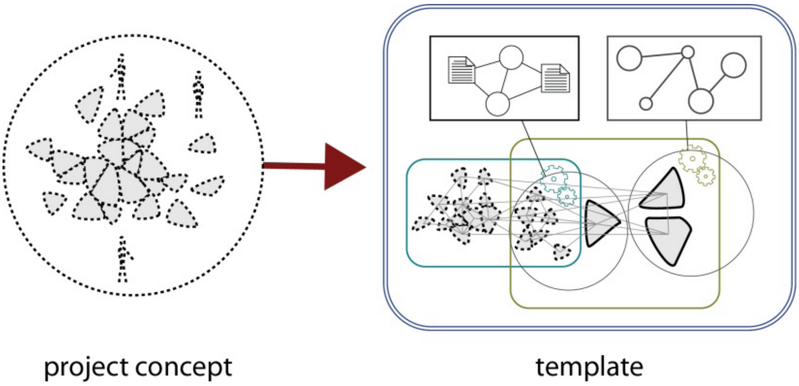


Fig.72. Initiation of an iA project by infusing a generic template with a focused project concept.

The initial development of an iA project in protoFRAME consequently focuses on embedding it in the context and gradual adaptation of the generic template to the specifics of that context. Simultaneously the development of the iA system can be instantly initiated and agents can be created. The template can include instruments, e.g. protoSWARM and protoNODE. In that case virtual agents can be immediately created in the protoSWARM environment and a set of protoNODE hardware can be instantly deployed in the physical environment, e.g. on the project site or in the test-laboratory.

At the outset of an iA project, a protoFRAME template can be extracted from an earlier developed project, or an earlier defined template can be reused. This permits evolution of templates across iA projects, as well as sharing best iA practices, patterns, techniques and instruments between projects by sharing protoFRAME templates. A large collection of templates can thus eventually be created. In such collection, specific templates would be provided for different sets of design challenges. Such collection could robustly evolve through a shared effort of designer community.

4. Forming protoFRAME templates

Conceptual design has shown to be a critical and time consuming phase in every iA system development process. During that phase all meta-models and models become initially defined. The conceptual phase is then followed by the iA system being set in motion. However, as discussed in section 1 and earlier throughout tracing of case studies in chapter III-VI, it is desirable that the iA system begins its operation simultaneously with the project development initiation, without the need to separate the design phase from the operation phase. Thus, the duration of the conceptual design phase should be reduced to minimum, while assuring the high quality and focus of the initial design concept.

The question inevitably faced when defining project templates involves the degree of specificity in respect to taxonomies of iA agents, relations and other aspects of developed systems. In established building industry practices, different classification systems for building components are used depending on the involved discipline and project location (examples include NL/SfB, CI/SfB, BSAB, CAWS, Masterformat, Uniformat and many others). It has been recognised in ISO STEP standards and following them IFC (industry foundation classes) that no single classification model can be imposed on building components, and IFC utilises a data model which allows multiple, concurrent classifications for every component, as well as translation between classifications[1]. This shows that even in established building industry, one shared classification standard is not feasible and even not desirable. Interactive architecture introduces a different set of problems that require different component classifications. In one iA project individual iA models can use different classifications of components, specific to the role of the model.

On the other hand a template needs to define the level of detail used in the iA systems. Therefore a generic overview of agents used in the iA system can be introduced in the template definition in order to indicate the level of detail of involved components.

4.1. Forming the initial iA project template

Formation and use of iA templates at the outset of each iA project is assumed as the main strategy for implementing protoFRAME. protoFRAME templates provide generic starting points for development of various iA projects. Each iA template includes a comprehensive set of metamodels and possibly a set of preliminary models describing the initial state of the iA system. Templates also provide a generic definition of employed system agents, with an explicit focus on the level of detail of these agents and general specification of agent types required to initiate the deployment of a working iA system. If needed, a preliminary set of generic iA system agents can also be included. Finally, also a set of iA instruments is also part of a template.

[1] 'IFC Overview Summary'., http://www.buildingsmart-tech.org/specifications/ifc-overview/ifc-overview-summary

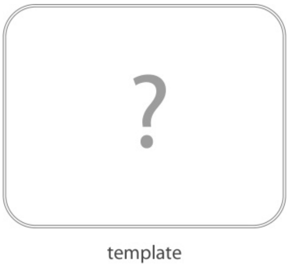


Fig.73. Ideogram: empty template

An iA template could be compared to a seed or an embryo of an iA project, however the fundamental difference is that a seed or an embryo contains the DNA of the organism to be developed, while the iA template is only a collection of means to initiate and facilitate the development of an "iA organism" (distantly comparable to a combination of RNA, enzymes, lipids an cell membrane). The role of the "DNA" in the initial project development is taken up by the design concept, which "infused" into the project iA template initiates the project development.

The goal of templates is to provide necessary means for project concepts to rapidly develop into preliminary but working iA systems. The early phase of this development does not have to take place in the target context of the project, being developed in a laboratory or in a virtual environment. Continuing the distant analogy to development of a living organism, this early phase shares some similarities to the development phase of a seed inside its shell, or an embryo developing in an egg or womb.

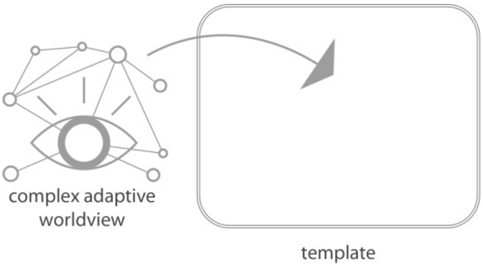


Fig.74. Ideogram: organisation of a template based on the complex adaptive worldview.

Brought up analogies to nature are very distant and cannot be taken literally, as the mechanisms perpetuating organism development in nature are ultimately very different in their operation from iA systems. However, the main difference between natural and iA systems is the inherent ability of iA systems to evolve during a single developmental cycle. It means that conceptual "blueprints" of an iA project, are not complete at the project outset, but increase in detail as the project develops. In case of negative validation during the project development time, the project concept guiding the development of the iA system at hand can modified, or even radically changed.

Ultimately, this process leads to complete project realisation from which modified or entirely new templates can be extracted to initiate development of new projects to come.

The following points provide three initial iA templates that are formulated based on the design research case studies traced in previous chapters. These iA templates are deliberately simplified, as they meant to be the first ones in the growing collection of differentiating, evolving and cross-breeding templates for complex adaptive interactive architecture. The templates are also introduced in a sequence of increasing complexity, where much of the initial template is included in the second one and the last template covers the widest range of the involved level of detail.

Interactive installations are currently the most common class of iA projects, limited by small scale factor, short lifespan and experimental character. Due to their relatively limited complexity, interactive installations can be seen as precursor systems for larger iA projects. Accordingly, traced iA prototypes have not extended beyond the scale of an interactive installation (see V.). Consequently, the initial protoFRAME template is geared towards iA systems on the scale and context of interactive installations. In later points of the section, extensions and modifications of such initial template in order to accommodate larger projects will be discussed.

The template for an interactive installation needs to be rapidly transformed into an experiential prototype and subsequently iteratively enhanced and improved. In case of an iA installation, context plays a relatively small role. An iA installation is usually not expected to be tightly integrated with its surroundings. However, the guiding design idea of an installation may require selective connections to project context.

An iA template needs to provide an initial set of networked system components and metamodels defining the comprehensive way in which these networked systems can be understood, analysed and further developed. Subsequent formulation of models needs to proceed alongside experiential prototyping

In traced design case study projects (see III.-V.), most realised projects (see V.) were on the scale and context of an interactive architectural installation. In retrospect, the main problem faced by designers of these installations was the long conceptual phase preceding the building phase and the lack of scalability of ultimately realised prototypes. The protoFRAME template needs thus to answer to both demands; it needs to allow rapid deployment of a working system of installation agents and it needs to allow perpetual development of that system into a concrete project, driven by the leading design concept.

4.2. iA system start-up components in a template

An iA installation can be assumed to focus mainly on human-scale interactions. Therefore, agents of an iA installation system can be presumed to have the scale to which humans can relate in 1:1 interactions (see V.3.1). That typically means object of the size from several centimetres to 1-2 meters. At the same time, virtual agents of an iA project are expected to be geared directly to serve practical purposes, as the iA system is set to be instantly realised in form of a working prototype. Therefore no pre-prototyping design phase needs to be accounted for. (Since observation from muscle, iPortals and interactive environments projects show clearly, that experiential prototyping provides critical influence on design concept development, not observed in projects such as MSc1 distributed faculty (III)).

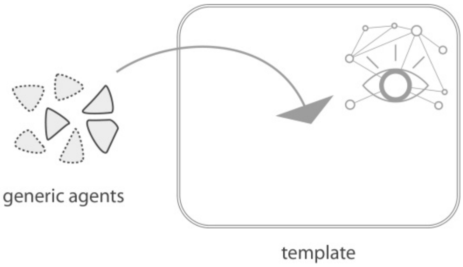


Fig.75. Ideogram: addition of generic agents to a template

The detailed taxonomy of iA project components can play a constraining role on designers if it favours certain established component types. On the other hand, a general classification of agents is required to facilitate fast deployment and development. What's more, the template can provide a set of operational physical agents, providing that they are provided in a generic way, allowing versatile paths of further development based on the project concept.

a) Physical building components

Physical components can be divided into:

- Active components; equipped with microcontrollers, sensors and actuators and can actively transform themselves and communicate with other agents.
- Passive building components; require external intervention to be modified.

Through external interventions passive agents can be potentially turned into active, while the opposite is also possible.

Building components can also be classified based on their geometry and structural performance into nodes, struts, surfaces or volumes (V.1.1), such classification, however, does not need to be enforced by the template as the choice of component geometry is directly related to the design concept, thus external to the template.

Interactive installation template needs to contain thus a) a kit of parts for realising networks of communicating with each other active components b) a kit of generic parts for initial prototyping for materialising active and passive physical components. In traced projects, initial installations rarely consisted of more than 5-10 independent active components (V.2.1), the number of components would gradually increase however as the projects developed. For the initial template it can be assumed that ability to deploy 10-40 active components should suffice at the outset of the project.

The following set has been chosen as the initial kit of building component parts and materials:

- 15 Arduino Fio boards with 800mA batteries
- 15 Seeduino Mega boards with power adapters and XBee radio shields
- 30 XBee series-1 wireless radio modules
- a collection of various plug-and-play sensors, including proximity sensors (IR, sonar), light sensors, force-sensitive resistors, buttons and potentiometers, several humidity and temperature sensors, and air pressure sensors
- a collection of diverse effectors, including LED lights, DMX-controlled stage light units, EL-wire and EL-foil lights with controllers, 15 linear electric actuators with forces 1200-6000N and sizes 60cm-1,5m, 15 electric motors, 30 high power h-bride circuits for driving electric motors and actuators, 15 relay switches for triggering other electric devices, 30 solenoid air valves, a mobile air compressor and 20m air tubing
- generic materials, including mdf, plywood, Perspex, eps and cardboard, screws and bolts, silicone rubber sheets

The above list of components allows realising a network of 10-30 active building components within 1-3 days. The diversity of sensors and effectors is aimed at encouraging diversification of projects built using this kit. At the same time, it is acknowledged that having certain material at hand, encourages designers to work with that material instead of choosing another, not directly accessible solution.

b) Virtual agents

Virtual iA system components are agents in an iA system without a specific physical body. They can be agents existing solely in designer's mind, on paper or in a virtual computer generated environment. In all cases the agent is valid when it causes change (thus its own agency) in any other agent of the iA system.

The use of virtual components may not seem essential at first when realising the iA installation. However, the design concept is in fact the first, rudimentary virtual agent without which the development of an iA system is not possible. The design concept can contain other virtual agents, either corresponding to physical agents that are to be created, or not being related to any spatial form (such as e.g. powerlines (III.2.1), economical agents, places and other).

The role of virtual agents is thus to complement the physical building components and perform tasks that physical components can't do. Thus, in this respect an iA installation with only one virtual agent being the design concept is possible, however inclusion of other virtual agents is inevitable and essential if increased complexity of the iA system is to be attained.

Similarly to building component agents, classifications of virtual agents are to be avoided. Nevertheless a general classification is needed as certain agent types need to be incorporated in the template. A generic taxonomy includes:

- Design concept agents; steering the development of the iA system, but not being integrally part of system operation. These agents cannot be standardised in respect to the way in which they define and implement the concept, as every design is fully unique and can imply intervention on any level of detail.
- Virtual component agents; correspond to physical agents before and possibly after their materialisation. There is a reciprocal relationship between virtual and physical component agents and that reciprocity needs to be maintained, i.e. the correspondence is not a representation, that is corresponding agents may share some features, while also having some unique features of their own that the corresponding agent does not have.
- Virtual spatial agents do not correspond to any material component and only function virtually in the system, while they do correspond to some position or area in the physical world. Such agent can be e.g. "space" which does not correspond to any specific material entity as air freely flows through it, but is defined implicitly by surrounding it components. In many project, specific virtual agents were created in order to achieve a desired experience, e.g. in Odyssey project (III.3.4) free virtual agents moving through the physical installation were driving the interaction with the users using the physical agents as intermediaries. Similarly, "weather" can be defined as a virtual agent which corresponds to the entire open space around the project, but does not have an explicit material form. Such weather agent can then have a number of sub-agents describing fields of temperature, humidity, wind etc.
- Virtual abstract agents do not directly correspond in any way to a location in the physical world or to any specific physical component. A "policy" is an example of such agent, which in principle applies generally to the entire project and has no spatial relationship.
- Group agents that define a group of other agents can be seen as either virtual or physical. A wall being a group of bricks, is logically a physical agent. However, in practice groups are often assemblies of both physical and virtual agents and their definition and any form of prescribed top-down agency that a group has on its content are also virtual (as there cannot be a microcontroller embedded in a group of physical agents). Therefore, as a convention, assemblies of agents are can be considered as virtual agents.

The distinction between such defined agent types is flexible and virtual agents can also move from one category to another. As an example, a building regulation can be created as a virtual abstract agents initially, but as project expands, that regulation can be converted to a virtual spatial agent, as it, for example, only applies to a section of the site.

c) Users

At the outset of research (II.1) it has been established that users are to be accounted for as agents within complex adaptive interactive architectural systems. The ambition of all realised design research case studies has been to directly involve users in the design and prototyping process (V.2.2).

However, the role of users in interactive installations differs from that in more generally approached architecture. Unlike in housing or office buildings, which people continuously use over longer periods and regularly, the users of iA installations are predominantly temporary visitors, meaning that their occupancy of the installation is usually short-lived, and they are unlikely to visit the installation regularly.

In traced projects the role of users has been changing throughout the project development. The following taxonomy reflects that change and can also be a guideline in planning of user involvement for project development.

- Visitors are actual target users of interactive installations. They are people who are usually not long-term involved with the installation and have not taken part in its development. They can have various social and cultural backgrounds, be of different age groups and sexes, have different intentions and needs towards the installation environment, including malicious intentions.

- Testers are actual users whose involvement can be moderated by the installation designers. They can be explicitly asked to perform certain tasks and they can be provided with additional information about the installation. Testers have been involved in traced Interactive Environments projects throughout their entire development cycle, which has been organised in sessions, and role of testers was taken up by children (V.2.2)

- Virtual users are virtual agents that correspond to actual people. This correspondence can have several forms. Virtual user can be an avatar controlled by an actual person (e.g. in d|e|form). In that case it allows testing of a system of virtual spatial agents before realisation of corresponding to them physical agents. Virtual users can also behave autonomously. In such cases (e.g. in paracity) they allow comprehensive testing of largely complex systems and reduce the time required to test with actual users and development of interfaces for avatars. However simulated virtual users are far-fetched simplifications and reductions comparing to actual users. Since societal, cultural, psychological, and individual factors play an important role of human behaviours at installation scale, autonomous (simulated) virtual user agents are to be avoided.

In order to form, manage and share virtual agents in the project team, to include virtual agents in the template and thus to also allow improvement of virtual agents across projects, models and instruments of realising and deploying virtual agents are necessary and will be introduced in detail in the following points.

4.3. Metamodels in a template

Traditional architectural models typically function as representations of a certain state of the designed building (VI.3.3.d). In traditional architecture there is stereotypically only one main state defined in the project with a limited number of very limited, local sub-states (such as turning on/off of lights, opening closing of doors, movement or no movement of an escalator etc.). An iA building can take an infinite number of states at all its levels of detail. Many states of an iA building need to be modelled in order to approach a comprehensive representation of that iA building, and that representation will have nevertheless limited temporal validity or limited level of detail validity. Therefore, state models on their own are not sufficient to comprehensively describe an iA building.

The second group of models required to in an iA project are models that show how various states of the developed iA system are interconnected, and thus, how the iA system changes over time, on all its levels of detail.

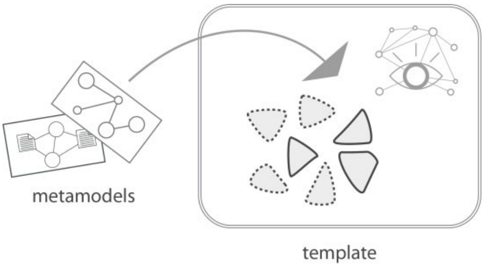


Fig.76. Ideogram: addition of metamodels to a template

The iA template needs thus to include metamodels defining rules for these two types of families of models to be formed by. Since the proposed initial iA template includes physical components, models of the initial system of these components that follow the above metamodels are also an integral part of the template.

Based on analysis of case study designs on the installation scale, following metamodels are defined in the initial iA installation template:

a) State models

- Multiple system states modelling. Multiple system states need to be easily modelled by the designers, in a way that does not constrain the design development. Simple notation for such model is a list of agents for each state with specification of parameters at a given state. Such lists are however hard to read and comprehensively comprehend by humans, and practically impossible to design with. Therefore an instrument is required to visualise agents and their states and to modify their parameters intuitively and efficiently.
- Relation modelling is principally a subset of a state model, where only relations between system components are modelled. In protoFRAME relations between agents are defined as properties of indidual agents (not e.g. entities separate in their own right).
- System organization modelling. System organisation is a subset of relations model, where explicit focus is placed on organisational relations between models, including semantic relations, proximity, etc.
- Users modelling is necessary in order to conceptualise their involvement in the iA processes at early stages and ultimately to relate to individual users in the realised project. A users model is a subset of a system state model, describing what users are part of that system at the given state. At early design phases those users can be virtual agents or personas, at later stages they are real persons. In that case a model is always a far reduction of who those persons are and needs to focus on identifying the qualities of those persons that are relevant to system operation.
- Context modelling involves creation of models that map system context, beyond system boundary.

- Interface modelling reduces the focus of the state employed state models to parts of the system involved in a concrete interaction process with a user or specific group of users. Interface models need to explicitly model the channels through which information is exchanged and feedback is collected.
- Electronic hardware implementation model is a specific type of a model which deals with defining how the electronic and electric components are embedded in iA components.
- Inventory is a list of all agents constituting a system in a given state, be it past, present or future of the system

b) Transformation models

Principally transformation models are models that relate various states of the system or its parts to each other over time. This can be done at any level of detail. On one hand states of the entire system can be connected to each other to describe project global development. On the other hand, more practically, states of invidual agents can be consequently causally connected to each other describing that component's behaviour and/or to states of other components describing interactions between them. Such models share many similarities with finite state machines[1], but are less rigorously approached.

However, an iA project consisting of hundreds of agents, where each agent can have thousands of possible states (even if approximation is used) gives practically endless possible states to be modelled and connected. Therefore only limited number of most representative states can be modelled (i.e. comprehensively represented) on the whole-system scale. A simple version of such model can be represented graphically as a flowchart. Project development models with tens or hundreds of states are impossible to depict in an easily legible manner. Consequently, the model of project development can be on higher complexity level formulated using a programming language. The choice of the programming language depends on many factors, one of which is the choice of the design instrument which needs to interpret chosen language. Among possible languages, graphical programming languages, such as virtools, MAX|MSP or Grasshopper 3D provide a combination of developing a flowchart with being able to deal with complexity of created representations. Consequently, transformation models can require translation in the process of design development, between simple flowchart (state machine) models which are suitable for describing limited state sequences to other models, which support working with systems of higher complexity and are supported by employed instruments. The following model types are initially included in the iA installation template:

- Project development modelling involves describing connections between various stages (states) of project development.
- Component behaviour modelling involves describing the operation of one component over time.
- Parametric component modelling describes the way in which all variables of a component are interrelated, reducing therefore the limit of states a component can have. Parametric component model complements the behaviour model.
- User activity modelling describes typical user activity patterns in the project, allowing anticipation of these patterns in the designed system. In that, a storyboard is a simple state narrative that describes such pattern in a comprehensible manner. User activity models complement interface state models.
- Procedure modelling needs to be employed where a specific, repetitive procedure is embedded in the iA system, such as fabrication procedure, or assembly procedure. Such procedures are simple lists of tasks to be executed in sequence to reach a goal.

[1] Ferdinand Wagner et al., *Modeling Software with Finite State Machines: A Practical Approach* (Auerbach Publications, 2006).

- Flow models are a family of models that represent flow of energy, matter of virtual entities (e.g. money) through the iA system over time. In that respect they define the flow mechanism as the rules of incurred transactions, rather than presenting a set of states in respect to that flow.

Described list of models is extensive, yet not exhaustive. On one hand two rudimentary metamodels defining the fundamental model types of state-models and transformation-models are theoretically sufficient to model any iA system. On the other hand, based on these two model types, many highly scpecialised models are created. In order to establish shared practices and accelerate development of these models, metamodels can be branched and specialised. The open question thus remains, to what level of detail should models be defined in metamodels and to what extent should they remain undefined and open for flexible interpretation?

The path chosen in protoFRAME is to initially leave the development of models unconstrained, enforcing only the division into state and transformation models. The extensible nature of protoFRAME allows for gradual increase of number and specialisation of included metamodels, based on results of applying various protoFRAME templates to consecutive projects.

On the other hand, the second aspect of metamodels involves their integration with insturments. Many models are specific for the instruments employed in the process; therefore instruments employed in the process may contain a highly specific metamodels, required for instrument implementation, independently of the general metamodel structure introduced by the template.

4.4. Instruments in a template

The fundamental class of instruments employed in studied projects, were instruments that "vertically" covered the degrees of abstraction of an iA project and permitted or facilitated development of a specific aspect (subsystem) of the project. Consequently an instrument can be seen as an integration of a metamodel, a medium for creation of models and facilitator for deployment of modelled systems in reality.

Numerous instruments were developed for case study projects as parts of protoKIT. All these instruments comply with protoFRAME meta-metamodel, since they were explicitly founded on the "multiagent worldview", where each part of the iA system is considered as an autonomous and unique agent. The protoKIT instruments were either developed from ground-up, or as extensions to existing commercially available products.

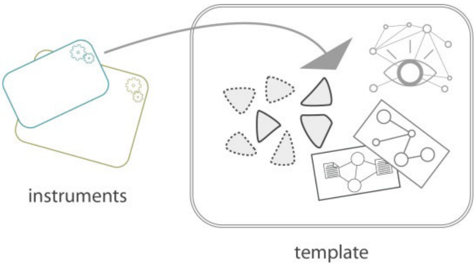


Fig.77. Ideogram: addition to instruments to a template

Studying installation prototypes developed in design research case studies leads to inclusion of the entire set of protoKIT instruments (see IV.2), including prototyping instruments (V.1) in the iA installation template.

a) Project organisation

Certain dichotomy spurs from the inclusion of instruments “vertically” across protoFRAME degrees of abstraction. Addition of an instruments means an addition of that instruments’ own set of detailed metamodels, models, agents and direct manifestation in project reality.

If metamodels included in the initial iA templates are purposefully left generic, the detailed metamodels of included instruments are the opposite. Although this may seem contradictory, such setup of protoFRAME is highly beneficial. The non-instrument metamodels are employed by designers in situations of certain “vagueness” where no instruments exist, or where freedom is required to explore various “out of the box” design possibilities. On the other hand, instruments are used for developing aspects of the system that are well known, where certain routines and patterns have been established and where innovativeness has to be channelled into details rather than into reconsidering the big picture.

In iA installations, observed project organisation has been horizontal. Even if hierarchy was imposed, teachers/project leaders would gradually take on tasks relating to lower levels of abstraction. Models were scarce, meta-models rarely shared, even among parallel projects. Instruments were used only if providing immediate benefit to designers/engineers.

The workflow in iA installations has been centred around the creation of an experiential prototype. Initially slow design process would significantly accelerate with the initiation of the prototype. Specialisations among project team members would develop in the course of the project. Changes of specialisations occurred in the process.

The lack of top-down direction has resulted in a number of investigated iA installations losing momentum and not being finished. In many cases designers were unable to reflect on the general purpose of the finished installation. For this reason, a pre-defined set of organisational rules proves to be a remedy and a parallel development of the project on higher abstraction layers is essential.

It can be expected that provision of proposed instruments, metamodels and template models can significantly accelerate and organise iA installation projects. Additionally, routines for high-abstraction design can support providing of larger scale and longer term contextualisation of projects.

b) Project instrument kit

The provision of instruments in the template is directly related to the generic system agents that are part of that template. Most agents listed in point 4.2 are to some extent dependent on one or more instruments.

Component type (see 4.2)	Instruments required (see IV.3, V.1)	Instrument-specific metamodel
Physical building components	protoFAB	G-code
	Rhinoceros3D/Grasshopper	Grasshopper visual scripting Rhinoceros geometry model
	protoTAG	protoTAG data structure
	protoSPACE	project description
	protoBASE	GUID based data structure
Active physical building components (including Arduino microcontrollers)	Arduino Boards and IDE Sensors and Effectors	Processing
Virtual component agents	protoSWARM	Virtools visual scripting language
	protoBASE	GUID based data structure

Template system components, corresponding required instruments and inherent to them metamodels

In the process of project development, instruments originally included in the template can be removed, replaced or changed if system development requires. Therefore, the critical quality of the included instruments is their ability to evolve and be modified “as they are being played” in the iA system. In this way, in their tight relationship to other parts of the iA system, instruments may need to evolve alongside other parts of the iA system.

c) Constraints

Inclusion of instruments tightly integrated in the template comes with many benefits. The protoFRAME structure becomes automatically enforced through the instruments and iA developers are provided with increased speed and quality of initial project development, while being able to freely shape the project into direction dictated by the project concept.

Nevertheless, instruments can provide constraints to project development. There is a risk that the development paths facilitated by the instruments will be more preferable to other paths, even if other factors would have indicated otherwise. Therefore it is of critical importance to organise project development in such way, in which models internal to instruments come secondary to instrument independent and more generic models.

The second main constraint comes from the use of proprietary instruments or instrument parts, which may limit the modification and consequent evolution of instruments, may partly hide involved metamodels from designers and may limit cross-evolution of templates across diverse projects, since accessible to proprietary instruments may be limited for other designers.

The last significant constraint is the cost of templates induced by instruments. This cost may be caused by software licenses, as well as, more significantly, by material costs. The material costs adding up to approximately 20000€ have been the main limiting factor in bringing the here described template to comprehensive early use and testing.

4.5. Differentiation of templates

The initial protoFRAME template is expected to rapidly branch into a number of specialised versions, following its first applications yet to come. The following points provide a hypothetical foresight into the expected further development and branching of the iA template.

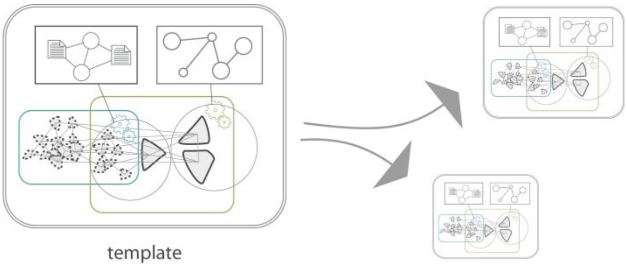


Fig.78. Ideogram: template differentiation

a) Interactive installation template

The initial iA template has been developed based on the iA system aimed at creation of interactive installations. However it has retained a rather generic nature, while at the same time assuming location of installation projects in the exhibition context.

The development of a more specialised iA installation template is aimed to change its target context. The exhibition context is very generic and is appropriate for situations when preliminary research or radically new concepts are developed and showcased. However, in order to develop iA systems further, contexts with a more specific functionality need to be found so that development and maintenance of a robust iA system can be financed.

While the initially assumed meaning of "installation" suggests the iA system to be an "art installation", the second meaning of the word "installation" in context of buildings means facilities added to a building shell (typically plumbing, electricity, fire protection, security etc.). Following this semantic duality, the iA installation template can be projected to be developed from an interactive exhibition installation to interactive building installation. Building extension can mean any form of addition to a typical, static architectural form. It can add a layer of interactive features to existing material building components, or it can add new components forming a physical extension of the building.

In that respect, the current iA template does not require far going modifications. However, the main concern is the robustness of the system, in respect to the protoFRAME template, meaning that more scalable and realiable solution for the microcontroller network and for the operation software needs to be developed.

In relation to the protoDECK 2.0 project and on-going cooperation with dr. Stefan Dulman and Andrei Pruteanu, first steps have been made to develop the iA installation template based on LPC Expresso microcontrollers, eLUA embedded software platform. This solution provides more reliable communication between nodes, as well as easy reprogramming of multiple nodes at once. The second major planned advancement is replacement of the current protoSWARM instrument, where Virtools development platform is to be replaced by open source libraries, allowing unconstrained further development of the platform.

b) Interactive building template

The main difference between an installation and a building is the concerned level of detail. An installation deals mostly with human scale phenomena and agents. Aggregations of physical components to higher-order entities, or people into higher-order social structures are rarely observable. Another distinct feature of installations is that they are not intended to be standalone. An art installation takes place in the context of an art gallery or another space. A building installation is added to the existing building structure.

The interactive building template is thus expected to develop from both the generic iA template and from the new iA installation template, into a template that can be used for initiation of entire, complex adaptive interactive building systems.

o System components

The fundamental difference between the initial iA template or the iA installation template and the iA building template lies in the initial consideration for system components.

• Physical components

The physical components in a building are more intricate, due to additional structural and isolative requirements, component size and durability.

Parts	Main components	super-components
All:	- nodes	- integrated components
-Sub-parts	- struts	- spatial clusters (rooms, courtyards)
Active components:	- surfaces	- functional aggregations (e.g. façade, floor, ceiling, wall, stairs)
-microcontroller	- volumes	- special components (e.g. elevator, escalator)
-sensors	- objects/devices	- furniture components
-effectors		

Examples of nested entities in an iA building

The initial iA template has dealt with nested parts in a building component. However treating the component sup-parts of main components has not been required, as the component has been assumed to be at the highest concerned level of detail for the iA installations. In iA buildings, the case is similar, however, on the other hand aggregations of building components form "super-components" which require to be recognised as iA agents at lower level of detail at which the building is dealt with.

- Virtual components

Similarly to building components, virtual components need to deal with larger scale and complexity that buildings have over installations. In these cases virtual components correspond to either aggregations of other virtual components, aggregations of virtual and material components or autonomous bodies that may have some kind of autonomy over their parts (e.g. social organisation as a whole has forms of power over its members). Consequently, many additional virtual components need to be taken into account. Their nature and complexity can be very high, and at the current preliminary level of the iA building template formation, they disdain clear classification.

- Users

The users of iA buildings largely differ from users of iA installations. The core group of users are the inhabitants of developed systems that use these systems persistently and might have partial ownership. Clients may or may not be inhabitants. They provide the initial request and financing for the development of the system. Maintainers are special kinds of users that perform system maintenance and can contribute do its further development. Maintainers can, but not have to be inhabitants

o Models-Metamodels

Rudimentary metamodels for iA building templates can be to a large extent shared with iA installation templates. However, due to higher degree of complexity in buildings, the level of detail needs to be extended to larger scale elements, while retaining the installation level of detail. The following list is only indicative and can be assumed as a starting point for model types that need to be taken into account. Over the course of project development, the following metamodels are expected to stratify based on feedback from corresponding models. As a result the next version of the iA building template can be based on a fully verified set of metamodels.

o Instruments

In the initial design phase an iA-building can be developed as an iA-installation and in that respect same instruments can be used. In later stages instruments used require more robustness and reliability. This is due to safety requirements, lack of designer supervision and maintenance, high usability requirements for inhabitants and significantly higher complexity of iA-building projects.

Appropriate set of instruments does not exist yet to date to fully fulfil these criteria. However through continuous developments of iA-installations such instruments can be gradually developed. Among available instruments, IFC standard and an IFC-compliant building information data exchange platform (e.g. bimserver) are required for large projects. Reliable interfaces, communication protocols, error checking, safety monitoring are required for full featured building projects. To achieve that reliability numerous advancements need to be made. However, experimental case study buildings could be realised with existing instruments and their performance verified in the realisation process.

o Project organisation

Architectural building process organisation involves significantly more participants and higher role and competence differentiation than it is the case with iA-installations. New approaches need to be investigated and verified in incorporation of installation engineers, structural engineers, contractors and maintainers of buildings into the development process. Initially traditional hierarchy of project leader/client – designers/engineers – users/inhabitants can be assumed, where within those three groups no further hierarchical separations are made. Based on practical experience, this model can be further adjusted or replaced.

o Project start-up kit

components	instruments	models
- 200 autonomous, working embedded nodes - collection of diverse plug-and-play sensors and actuators - set of generic physical prototyping material	- knowledge exchange wiki - online inventory - simulator - programming GUI - fabrication tool chain - parametric design software, enhancements for state model integration and exchange	- initial organisation model - library of sample models

Deployment components

The process of deployment of interactive building template can be partly anticipated based on the d|e|form project. In that case the iA system has been envisioned to be provided to site inhabitants as a ready to use package, providing a possibility to build a variety of simple building structures directly by the inhabitants, similarly to a process of making a construction out of children's toy blocks, however, with the difference that new, customized, blocks could be fabricated on the fly. Along the process, inhabitants with help of designers and engineers would develop more specialised and differentiated component types.

In such approach, however, the buildings have shown to develop without conceptual integrity, resulting in chaotic structures. Therefore, the involvement of users in deployment of a template may be postponed, as the template is being adapted and as integrative rules are implemented by designers based on the design concept in the preliminary development phase.

c) Interactive urban development template

Interactive architecture on urban scale poses a different class of challenges than in case of iA-installations and iA-buildings. Urban systems are highly dependent on complex and dynamic contexts and operate within overlapping networks of traffic, communication, society, environment and others. The level of detail of urban iA systems is wider than this of an iA building. On one hand the proliferation of mobile devices throughout urban space and their growing role as interfaces not only among users but also between people and artificial "things", make them fundamental iA instruments for urban projects. On the other hand, dealing with the urban problematic requires addressing entire city block and areas, as large scale agents in the system, that contain several degrees of nested smaller scale agents, such as individual buildings, dwellings or rooms.

Although iA can ultimately apply to urban planning scale, currently that scale can be considered most distant from direct applications. In urban design the key distinction from architecture is the public character of designed spaces and large numbers of participants.

Case study projects such as the 751 city showed great potential for interactivity on urban design scale. iA-urban can be seen on three scales. It can be design of interactive spaces in-between buildings (existing or new). It can be aggregation of iA-buildings and in-between spaces, yet it can also be a bottom-up intervention into existing city on the urban scale. Each of these project classes requires a different approach, yet can initially share the same template, which can be in the future differentiated.

o Components

• Physical components

Selection of physical components in urban design projects depends on the scale of the project and on its nature. The project class of iA-urban intervention is characterised by using small scale components. These components can be deployed on any area size. iA-public space project class, deals holistically with a project of a public scale, thus it involves components that are aggregations of smaller ones, which can include iA-building or iA installation components. Eventually iA-urban plan projects deal with systems of entire buildings, streets, parks etc. Projects such as Manhal Oasis proved viability of iA approaches on that scale. iA urban projects allow to mix outlined project classes and transform them between each other.

Main components (iA urban intervention)	super-components (iA public space)	Super^2-components (iA urban plan)
- nodes - struts - surfaces - volumes - objects/devices - vehicles	- functional aggregations (e.g. pavilion, kiosk, playground, bus stop) - spatial clusters (spots, courtyards)	- buildings - streets - parks - squares - water bodies

Nested iA urban project compoentns

o Models-Metamodels

Metamodels employed in urban planning traditionally allow more flexibility for implementation. Land-use plan (dutch "bestemmingsplan") defines the plan as a set of spatial regulations, rather than specific physical design. Such land-use plans can be adopted to accommodate adaptive stractures. On smaller scales same metamodels as in iA buildings and iA installations can be used. However, due to specificity of the urban context, additional metamodels need to regulate modelling of different forms of traffic, public and commercial services, public-private gradients and ownership. Flow models for users, vehicles, money, material or energy and related transaction models have proven to be of use in urban design. Social and cultural models and user/inhabitant participation models are also required.

iA-installation and iA-building models	iA-urban-specific models
- system state models (IFC compliant)	- land use models with adaptation-oriented notation
- system organization and grouping models	
- user and group activity pattern models	
- persona and user group models	- social organisation models, crowd-sourcing models
- scenarios/storyboards	
- parametric component models	
- fabrication and mass customization procedures and organisation models	
- component behaviour model (in implementation-independent notation)	
- electronic hardware implementation model	
- interface models (between active components, humans)	
- short-term development and long-term development strategy models (including creation and removal of components and shifting of boundary)	
- relation model (including semantic relations between components)	- social relationship model
- inventory (inventory of all system components, present, past and future)	
- financial planning, strategy and maintenance models	
	- legal and ownership models
- context models (mapping of system context, beyond system boundary, social, cultural, political, infrastructure, etc.)	- context development models
- user feedback models (models for incorporating user feedback into development process) and social participation models	
- building evolution model	
	- energy/matter/money flow models
	- infrastructure models

Extension of iA-installation and iA-building model set with iA urban-specific models

o Instruments

iA urban design projects are expected to involve much more extensive and persistent participation of inhabitants than it may be the case with iA-installations or iA-buildings. In many cases the success in involvement of inhabitants may determine the success of the project. For this reason the role of the instruments to engage inhabitants has additional relevance for iA-urban projects.

Additionally, the highly complex nature of urban design challenges, requires additional simulation of proposed systems before their deployment. This is needed to identify and minimize negative effects that deployed systems may produce in an emergent manner.

A specific kind of an instrument is a legal instrument, which through local law (where e.g. the approved land use plan, becomes local law) allows enforcement of certain models on reality.

Additional instruments for iA-urban projects include:

- Online massive-collaboration platform
- Collaborative decision making facility
- Simulations
- Legal instruments

o Project organisation

iA-urban projects organisation is difficult to generically structure. Urban planning projects are traditionally executed in a top-down manner; however the current trend is to increase the role of inhabitants in such projects. iA-urban projects have the potential to empower this trend. A balance between top-down and bottom-up design needs to found through further case study projects.

o Project start-up kit

A combination of iA-installation and iA-building start-up kits can be used, depending on initial project scope. Additionally, the following are provided:

components	instruments	models
- 200 autonomous, working embedded nodes - collection of diverse plug-and-play sensors and actuators - set of generic physical prototyping material	- knowledge exchange wiki - online inventory - simulator - programming GUI - fabrication tool chain - parametric design software, enhancements for state model integration and exchange - public online interface to the project	- initial organisation model - library of sample models - initial land-use plan

Project start-up kit in urban development template

o Deployment

The deployment of protoFRAME urban templates can be partly illustrated with a class of urban-scale case study research projects traced in section III.7. In all those projects parts of the indicated template have been provided at the outset of project development. Consequently, it has been observed that clearly defined guidelines for models and metamodels on urban-scale were critical for project success. The quality of final results has observably increased proportionally to the clarity of imposed design rules and the availability of design instruments. Nevertheless, in none of the traced projects the connection to the actual project context has been significantly limited. It can be expected that further development and introduction to the iA urban template of instruments for connection to context inhabitants and project users alongside development of methods for increased participant involvement in design processes can bridge the gap between the virtual designs on that scale and their deployment.

4.6. Towards formation of iA design and realisation methods

A "method" can be defined as a systematic procedure aimed at attaining a specific goal. Consequently, an iA design and realisation method is an intricate systematic procedure aimed at creation of complex adaptive interactive architectural environment. protoFRAME delivers a highly specific structure for developing such methods. Project initiation using discussed to this point iA templates in combination with employment of specific models, instruments, project organisation and adds up to comprehensive method for design and realisation of iA.

Out of traced design research case study projects, a number of methods emerge. These methods are inherently tied to the chain of degrees of abstraction. A method involves a specific worldview and metamodels for formulation of method-specific models. It also applies to a specific subset of system's agents. Consequently, an iA method can also be inherently tied to a specific instrument, which might be necessary to attain creation of certain subsets of an iA subsystem and its model.

Throughout development of iA projects, various methods can be applied. These methods can provide a link between project similar to templates. The main difference being that a template provides a starting point for a project, while a method provides a strategy to be applied during project development. Yet, similarly to templates, methods can be evolve across projects.

In this view, methods are systematised procedures. Designers can employ these procedures in order to increase the speed and quality of the iA development process. Since in many projects inhabitants and other human agents take up roles of designers, methods can be applied throughout all aspects of the development and operation processes of iA

In the formulation and development of iA methods, the key problem is the validation and verification of the methods, in conjunction with their documentation and reuse in other projects. Tracing of actors across systems can be seen as a useful way for such validation.

4.7. Conclusion

The section has started by providing an overview of the finalisation phase in the formation of the initial iA project template. Consequently, a speculative process of further differentiation of that template into several specialised templates has been discussed. As a result, emergence and development of methods in conjunction with employment of templates has been preliminarily discussed.

iA templates structured in accordance with protoFRAME appear as promising means for acceleration of iA development and as means to fuel cross-project evolution. Resulting methods supplement that evolution by providing horizontal links between projects throughout their development.

5. Future challenges

protoFRAME provides an extensible structure and organisational foundation for developing iA projects. As shown, extensive research needs to be further conducted in order to advance iA agents, models, conventions, methods, instruments and templates and in result bring iA to practical applications. This research need to be conducted on all fronts discussed in this chapter, in conjunction with continuous validation of realised iA projects in diverse contexts, ranging from spatial to social and cultural. In this process the main challenges are various kinds of cultural adoption of iA design which needs to be followed by assembling new communities of iA designers and users. iA systems need to prove scalable and their response to human spatial needs, has to be empirically proven and perfected.

5.1. Evolution of agents: models: conventions: methods: instruments: templates

The previous sections have gradually introduced protoFRAME, and notions and processes that protoFRAME introduces and stimulates. It can be observed that a series of dependencies occurred in the described process of formation of protoFRAME. iA agents are contained in iA models. iA models require conventions (metamodels) for coherence within and across projects. From such coherence repeatable methods emerge, which can be improved over time. Instruments can additionally facilitate and advance generically encountered aspects of iA development. iA templates ultimately bring together instruments, methods, metamodels, models and initial system agents to accelerate and facilitate initial iA project development.

In such chain of dependence, evolution can be applied to all links in that chain. iA agents, models, conventions, methods, instruments, and templates evolve across projects and mutually influence one another.

a) Agents

The evolution of agents is the most observable one when tracing iA projects. In traced case study projects development of agents occurred at every scale and in every project. When generic system agents are introduced with a project template, and when a new or improved template is extracted from the developed project, such agent development process is brought to a full circle.

The evolution of agents can be twofold. It happens across projects, upon the evaluation of one development cycle, before starting the next. It can also happen during one development cycle by structurally changing the "DNA" of the agent in a way earlier not anticipated. Such in-process evolution is critical for the operation of iA and self-improvement of a developing interactive iA system. protoFRAME facilitates both forms of agent evolutions at different degrees of abstraction.

b) Models

Models can represent any part of an iA system in an instance of time or across a certain timespan. This means that also mechanisms of evolution of agents can be modelled. In this way evolution of agents within and across systems can be implemented and ensured.

However, such mechanisms of evolution may also require improvement over time. The improvement of these mechanisms happens within established rules and conventions, also referred to as metamodels.

c) Conventions

Modelling conventions provide ontological canvas for formation of entire families of models. If these conventions prove insufficient for advancement of the project, they can be modified. In this way, a slow-paced evolution of metamodels occurs, being forced to make a step when evolution of modes and consequently agents in a certain direction is no longer possible.

On the other hand, metamodels can also be changed more radically, to stimulate evolutionary "jumps" in situation when local optimum of a system is reached, but it is expected, that globally, better optima exist. In such situations, a radical change of a metamodel, enforces reformulation of models and may permit evolution of agents in new directions, or formation of entirely new agents.

d) Design methods

Design methods can be seen as a specific form of conventions defining repetitive processes in which humans influence the development and evolution of iA systems. Once a certain process is repeated several times with success, it crystallises into a method. That method can be consequently gradually improved and refined. The process of refinement and improvement of a method can be considered as a form of method evolution.

e) Instruments

The iA instruments combine agents, models and metamodels and stimulate emergence of methods specific for the instruments. Instruments are gradually improved over time and in this they evolve, as traced in chapters V and VI. The evolution of an iA instrument can involve only the internal mechanisms of the instrument, or it can also involve changing the features of bound to the instrument agents, models, metamodels and methods.

f) Templates

Continuous improvement of agents, models, metamodels, methods and instruments drives development of new templates. Over time templates accumulate and can be cross-bred for further improvement. A template exchange platform is essential to permit such exchange.

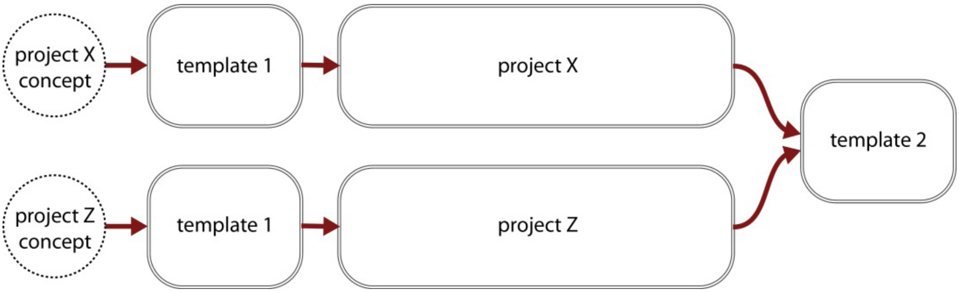


Fig.79. The basic mechanism driving the evolution of templates across projects

g) Conclusion

As shown, various aspects of evolution can be traced in iA projects. It is the sum of all those aspects that provides the high potential for iA projects to proliferate and specialise, while developing forms, affordances and spatial experiences enriching our habitats and fuelling their sustainable growth and well-being of their inhabitants.

At the same time, these processes are beyond human capacity to comprehend. protoFRAME provides a structure for organisation of the processes in a way that allows their dissection and layered development, without constraining any of the identified aspects of iA evolution.

5.2. Stimulating cultural adoption of iA

Interactive architecture in general, and complex adaptive interactive architecture more specifically, has shown to have no cultural precedents. Because of this lack, users and designers alike have shown difficulty in comprehending iA, being able to deal with its complexity and driving its development and evolution.

a) Designers, engineers, experts

The cultural adoption of protoFRAME among designers, engineers and experts mostly relates to the development of 2nd degree of abstraction. It involves formulation of metamodels that would be accepted by these communities and applied, in models and used instruments. As use case research has shown, these metamodels are unlikely to succeed if top-down imposed on the design and engineering community. Therefore they need to be developed with that community.

b) Users, participants

The other aspect of cultural adoption challenge is the acceptance of users. It remains unknown, what iA features will be desirable by its users, how different use scenarios will develop in daily reality. It is likely that some iA features will be entirely rejected by its end users, while others will find wide acceptance. It can also be expected that cultural acceptance will be accomplished after initial cultural rejection.

c) Clients, market

Yet, the most critical aspect of the cultural adoption is this of the market and clients, who in architectural praxis are often not the same as end users. iA needs to be proven as commercially viable through gradually increasing complexity and scale of iA projects.

d) Conclusion

protoFRAME in itself, remains too complex for most designers and users to swiftly manouver between its degrees of abstractions. However, the structure it provides allows designers and users alike to neglect higher levels of abstraction and focus on the easily comprehensible 1st and 2nd DOA. In this way highly abstract concepts can be removed from daily practice and only dealt with if explicitly required for an evolutionary leap.

5.3. Building the iA community

The cultural adoption of protoFRAME among developers and users goes hand in hand with building the community of specialised iA experts and adopting users, who need to take it upon them to advance development of iA as a field, realisation and inhabitation of iA projects.

The iA community is ultimately the key factor of iA's success or failure. The growing and vibrant community is bound to drive iA development and proliferation, while without the critical mass of developers and users alike, iA's existence cannot be mandated.

It is the hope of the author that protoFRAME alongside its templates, instruments, methods, metamodels, models and diverse prototypical agents will continue to fuel the iA community and lead to the increasing adoption of iA.

5.4. Improving scalability of iA projects

The distributed systems approach to iA, resulting in the branch of iA referred to as "complex adaptive interactive architecture" has been chosen due to its scalability. Case study practice has shown that iA systems structured in a centralised way, both in terms of design and technology cannot be scaled. Distributed approach needs shows great promises, yet new challenges are expected with scaling distributed systems to large building and urban applications.

5.5. Validation of iA projects

Ultimately, only realised projects can prove or disprove the rationale behind creation of iA and the applicability of protoFRAME and its constituents. The complexity of architecture, interwoven with all aspects of human life and society can only be evaluated holistically. Shall iA prove to be capable of becoming part of human culture, it can enhance and transform that culture in a vast multitude of unforeseeable ways.

VIII. Conclusions

Summary:

The final chapter in a compact form re-iterates through key findings and conclusions of the dissertation and reflects them back on the research framework initially formulated in chapter I.

5.1. Response to research questions

a) “What are the characteristics and features of the process of developing an interactive building as a dynamic complex adaptive interactive system?”

- The processes of development of complex adaptive iA are continuous and have no easily definable start- and end-points.
- The complex adaptive iA processes are characterised by simultaneous development (growth and differentiation of components) and evolution (introduction, evaluation and removal of new features and types in component aggregations) [1].
- There is no explicit distinction between design and operation of an iA system.
- Like any system, a complex adaptive iA system has a boundary and a finite number of components.
- Users of a complex adaptive iA building are inherent constituents of the system of that building.
- The role of designers and experts in the complex adaptive iA process is to regulate it, steer its development, introduce components and set rules of their operation. It is not to determine the spatial outcome of the interactive architectural process.
- A complex adaptive iA system can contain non-adaptive and/or non-autonomous components.
- Effort is required to keep developed complex adaptive iA processes in balance, limiting their tendency towards high entropy states by means of negative feedback loops.[2]

b) “What taxonomies and organisational rules are required for the development process of complex adaptive interactive architecture to unfold and sustain itself?

- Flat organisational taxonomies with ad-hoc “organic” grouping of system components and resulting flexible semi-lattice (non-hierarchical) organisation of system components have been evaluated to deliver preferred organisational structures in iA systems[3].
- In studied iA systems, qualities such as system transparency, extensibility, openness (exchange of matter and information through system boundary) have been identified as key to successful iA process development[4].

[1] See section VII.2.2

[2] For reference regarding all listed point see sections III.4-5, IV.8, V.4, VI.4, VII.2

[3] See project tracings in chapter IV

[4] For summary see section VII.2

- Heterogeneity of building components increases comprehensive adaptation and system evolution in large scale complex adaptive iA systems[1].
- Local connectivity among agents (human and non-human)[2] is essential for operation, adaptation and evolution in iA networks[3].
- Shared communication conventions and rules of engagement need to be determined and shared for an iA system to function and for cross-project development[4].
- Open knowledge exchange has the potential to extensively accelerate the cross-project advancement of iA systems in general, and complex adaptive iA systems specifically.[5]

c) "What technological enablers are required in development of complex adaptive iA?

- Instrument(s) for virtual deployment of early-stage iA systems (ontologically and functionally different from modelling or simulation environments)[6] are required in early development stages of complex adaptive iA systems and can be used to extend operational iA systems in later development stages.
- Embedded platform for operation of active physical iA components[7] is required to permit prototyping and deployment of iA systems.
- Network platform for active physical and virtual components[8] is required to provide an ability of non-human system components to communicate and interact among each other.
- Knowledge exchange platform[9] is required to accelerate development of practical knowledge in the emerging field of iA.

5.2. Response to research objectives

a) To further validate the largely distributed approach towards creation of interactive architecture.

Aspects of the distributed approach have been validated in over thirty case study projects[10]. In all projects the added benefits of non-hierarchical distributed project structure were apparent. However, it has been clearly observed that such approach provides a wide range of new problems and challenges, further discussed in section 5.4. Moreover, direct correlation has been oserved between the largely distributed approach and formation of interactive architectures as complex adaptive systems.[11]

[1] See sections IV.6-7

[2] See section VI.2

[3] See section VI.2

[4] See sections V.4, VI.4

[5] See sections V.3.9 VII.2.3, VII.4.3

[6] See section V.2, section V.3.2

[7] See section VI.1

[8] See section V.3.10, VI.2

[9] See sections V.3.7-9, VII.4-5

[10] See chapters IV-VI

[11] See sections VII.4.6.

b) To rigorously formulate an iA development framework allowing creation of comprehensive iA projects, providing the foundation for future iA methodologies and enabling the execution of the first research objective.

protoFRAME; the framework for interactive architecture formulated in chapter VII, integrates best practices, approaches, techniques, strategies and methods employed in studied design research experiments described in chapters IV-VI. Yet, above all, it organises iA projects using the "generic worldview" model, which is based on complex adaptive systems ontology. Although aspects of proposed generic worldview are not new to architectural theory, they are not common in architectural praxis. Consequently, protoFRAME also provides a set of practical means for organising and working with projects based on such worldview and to facilitate its practical implementation.

As a result, the value of protoFRAME lies mostly in its capacity to organise iA projects in a consistent manner, yet without imposing any detailed standards and constraints. In this way, within the general ontology proposed by protoFRAME, an infinite number and variation of metamodels can be created. These metamodels correspond to "specific worldviews" of individual project participants or groups thereof, as well as to highly specific worldviews induced by implementation of technologies. In this context, the role of protoFRAME is to link specific worldviews with the generic worldview that binds the iA project as a whole and allows relating iA projects to one another.

The diversity of metamodels and corresponding domain-specific worldviews present in an iA project, promotes diversity and multi-objective development and efficiency in iA systems. Concurrently, protoFRAME as a whole guarantees that such plurality does not become destructive to the system by providing a shared structure for deployment and interoperability of produced models and the actual operation of rich and heterogeneous iA systems, alongside support of best practice models and development of design patterns.

The constituents of protoFRAME have been individually tested and validated in presented design case study research projects. protoFRAME seen as assembly of those constituents is therefore proven to be an efficient solution for future iA developments. At the same time, the extensible nature of protoFRAME guarantees that through the process of its future employment improvements will be made, increasing its efficiency and in a bottom-up and crowd-sourced manner "breeding" and "cross-breeding" its future, domain-specific variations.

5.3. Evaluation of hypotheses

Initially assumed hypothesis stated that the "degree and quality of architectural adaptation can be significantly improved by replacing traditionally centralised and hierarchical organisation of architectural systems by a largely distributed, open and extensible one, leading to foundation of new methodologies for interactive architecture." It has been concluded that this hypothesis cannot be completely proven, although background research has initially indicated its validity. Consequently, the design research case study experiments have illustrated numerous aspects of adaptations made possible or facilitated by largely distributed approach to formation of iA systems, leading to creation of complex adaptive interactive architecture, which further validates the initial hypothesis. Among those are:

- Resilience of distributed systems from damage and errors[1].
- Ability for local adaptation while retaining global performance[2].

[1] See sections IV.4-7

[2] See sections IV.3,V.3.2,VI.2-4

- Ability to develop and evolve[1].

Nevertheless, since no exact metric can be employed to evaluate multi-faceted architectural adaptation and since no full-featured case study iA projects exist that could be used to comprehensively evaluate the degree and quality of actual adaptation, no final conclusions can be drawn in respect to the actual performance of complex adaptive iA systems resulting from the largely distributed approach to iA development.

In order to facilitate future research in the domain of iA the protoFRAME framework has been proposed and formulated throughout the dissertation as summarized in the previous section. This framework can be seen as an "inversely constructed hypothesis", as it has been postulated not *a priori*, but has been constructed gradually through the design research experiments.

protoFRAME is consequently assembled out of most successful ontologies, patterns and practices encountered in studied projects[2]. The extensible manner in which it has been defined allows for its further adjustment, while guaranteeing a shared frame of ontological and organizational reference for future projects.

Aspects of protoFRAME have been gradually introduced, evaluated and adjusted in design research case study projects. This process has not been linear, since various kinds of projects were employed in parallel to validate different aspects of protoFRAME. The chronology, however, has been maintained within sections of chapters IV-VI. Following that chronology it has been clearly observed and evaluated by external experts that the speed and comprehensiveness of developed iA projects has significantly increased with the introduction of protoFRAME aspects. The metric used to evaluate this progress can be the number of adaptive components and the number of delivered project scenarios. Nevertheless, only actual commercial deployment of iA projects can ultimately prove protoFRAME's validity and allow to compare and select its most successful variants for further development.

5.4. Facing the problems of iA

a) Lack of comprehensive reference projects

Studied case study projects, despite their broad scope and diversity did not individually deliver comprehensive examples of iA. However, in aggregation they did address many problems faced by complex adaptive iA systems. By summing up individual features of studied projects a better outlook on the comprehensive nature of iA systems has been gained. Nevertheless, the combination of these aspects in a single project still needs to be performed and verified in future.

b) Scalability

Following the initial expectations, scalability of iA systems has presented itself as a significant bottleneck in studied cases. Both in respect to virtual and physical prototypes, the number of adaptive iA agents has been constrained by used technology. However, this limitation existed mostly in situations where dependence on centralised control was required. Advancement of instruments such as protoSWARM[3] or protoNET[4] promises to provide robust partial to problems of scalability in iA systems in the future.

[1] See sections II.5.6,VI.4

[2] See chapters IV-VI

[3] See section V.3.2

[4] See section VI.2.1

c) Society-embedded constraints

The general perception of the domain of interactive architecture remains a significant problem in respect to future research. Nevertheless, public response to prototypes realised within the scope of the dissertation has been largely positive and delivered a greater understanding of iA potential among non-specialists as well as designers and engineers. It can thus be assumed that further advancement of iA case studies is likely to improve the interest and eventual demand for iA.

The problems related to financial and legal models present in architectural design, development and inhabitation practice remain a separate category, which has not been thoroughly investigated in this dissertation and which require to be addressed before iA can be brought to commercial applications.

d) Lack of rules of conduct

Based on its initial applications, protoFRAME promises to be a solid foundation for future methodologies for creation of complex adaptive interactive architecture. Methods applied to performed design research case studies require further validation, yet can already serve as a basis for initiation and rule-set definition of comprehensive complex adaptive iA projects.

5.5. Future of iA

a) Towards realization of complex adaptive iA

The future challenge for development of complex adaptive iA is deployment of iA processes in broader context. New cultural and societal models, financing models, ownership models are bound to appear alongside iA. The exact nature of these models cannot be predicted at this stage of iA development, however tendencies towards crowd-sourcing, micro-financing or use of open source solutions can already be observed. At the same time legal and safety-related factors are expected to pose significant constraints on the development of iA in the near future.

It can be hoped that employment and advancement of protoFRAME can provide substantial help in organisation of iA processes and in the future regulation of these processes to ensure safety of its inhabitants, as well as predictability in respect to global performance of otherwise emergent adaptive iA systems.

It is not herewith claimed that protoFRAME provides an ultimate iA framework. On the contrary, protoFRAME is considered to be a still raw development that needs significant further refinement. However, such refinement can only take place through application of protoFRAME in increasingly more complex, larger-scale and longer-timeframe case study projects.

The design research case studies investigated in this dissertation have been highly constrained. They have been performed in partly isolated environments, without in-depth consideration for market forces, ownership models, cultural preferences, social challenges or many legal and safety constraints that regular buildings have to conform with. It is therefore acknowledged that next steps of the research trajectory of complex adaptive interactive architecture development have to be oriented towards realisation of iA case study projects in more comprehensive real-world scenarios. Such case studies are expected to provide continuous feedback to protoFRAME and guide its iterative improvement and diversification.

At the same time, technological solutions utilised in most of the investigated design research projects have also demonstrated highly limited scalability. Both software and hardware used in realisation of studied experimental installations has been aimed to deliver proof-of-concept operation, while allowing rapid and extensive adjustability and modification. Conversely, in full-featured building applications long-term operation and reliability are required. This means that substantial further effort is required to perform the shift from experimental to real-world iA applications. What's more, tested installations were built at small-pavilion scale at most. Projects on whole-building scale require interoperability of components in numbers that are several orders of magnitude larger. Although the assumed distributed approach to formation of interactive systems is in its principle fully scalable, many technological solutions employed ad-hoc within protoKIT are not. Additionally, in real-world applications, cost, maintenance and security are issues of primary concern, which have not been explicitly addressed and validated in this research.

The two prospective research areas that emerge from the above observations involve 1) up-scaling and broadly contextualising iA case studies 2) technological improvement of materials, hardware and software for design and operation of complex adaptive interactive buildings. Both of these research areas can largely benefit from protoFRAME. Its application promotes heterogeneity of future projects, while permitting exchange of new and formation of improved iA solutions on methodological, technological and societal grounds. Multiple new materials, types of adaptive building components and supporting technologies are expected to emerge across all projects utilising protoFRAME, while globally, their reliability and efficiency is expected to consistently increase. At the same time new societal, cultural, ownership, financing and legal models need to be developed to permit inhabitation of future interactive architecture.

b) Complex adaptive future of architecture

When attempting to summarise and formulate an outlook on new possibilities resulting from the presented research, a larger question about the future of architecture comes to the foreground. What is going to be the nature of future architecture? Does development of interactive architecture mean that all architecture will become dynamic, interactive and adaptive?

This question disdains easy answers, as it can be claimed that much of the present-day architecture already is dynamic, interactive and adaptive. Yet, the speed and efficiency of adaptations in present-day architecture are low. Development of interactive architecture means thus that such speed and efficiency can be largely increased. The revolution does not lay therefore in interactive architecture being a mysterious new kind of architecture, but in shifting mode of perception of any architecture as being complex and adaptive, and designing and using it accordingly.

At the same time, the above statement does not imply that fast spatial adaptation will necessarily become ubiquitous. On the contrary, many scenarios can be imagined where rapid transformation of architecture is not and never will be needed, where stability, and continuity of spatial reference is desired instead. Yet even in such cases, complex adaptive view on architecture retains its validity and relevance. Preservation of buildings requires often as much, if not more effort as creation of new ones. For example, buildings need to be actively repaired and maintained. Thus, in order to uphold some of their features, many of their parts need to be adapted, so that they can sustain their functionality and usability in the world that changes around them.

In this way, development of complex adaptive interactive architecture does not need to entail a vision of future cities filled with dynamically transforming or moving buildings. Conversely, it can equally point us towards development of cities which are seemingly static, but which actively optimize themselves by preserving parts which are efficient and adapting and transforming parts which do fail, while allowing for dynamic spatial transformations only where they are explicitly desirable by city's inhabitants. Consequently, a balance is bound to be found between dynamism, transformation, adaptation and comfort, familiarity, safety, and confidence in buildings and cities.

Interactive architecture should also not be reduced to a technical feature of built environments. Its realisation is bound to have a profound influence on our culture and society, our way of thinking and our values. Interactive architecture can deliver new means for association of memories and history with space, and attribution of value to architectural systems, which are by definition in a process of perpetual motion.

What all possible scenarios of interactive architectural future have in common, is the increased role of inhabitants in formation and transformation of architectural habitats. Beyond doubt, the future of architecture is participatory, focused on enabling people to form and improve the spaces in which they live their lives.

Clearly, the future of architecture as sketched above also requires rethinking the role of an architect. The very concept of interactive architecture implies that an architect cannot be seen as sole creator of a building. However, it should not be assumed that interactive buildings can be created only by their inhabitants. There will always be need for moderators, integrators and "game masters" that design and set rules for games that multiple inhabitants and stakeholders will continuously play in the ever adaptive complex architectural habitats. Thus, as Mark Shepard writes, "The profession has a decision to make. Either it can cede the role of being the primary agent in shaping our spatial experience of the city to the designers and engineers of (embedded, mobile and pervasive ibid.) technologies, or it can shed its interdisciplinary anxieties regarding the purview of its practice and take part in shaping these technologies."[1]

[1] Mark Shepard, ed., *Sentient City: Ubiquitous Computing, Architecture, and the Future of Urban Space* (New York: The Architectural League of New York, 2011).

Summary

The first chapter presents a compact summary of background research in the domain of interactive and adaptive architecture. Based on this investigation, the framework of the research is concisely set forth, providing the canvas on which the dissertation further unfolds in the later chapters.

The second chapter comprehensively discusses the rationale of adaptation of buildings and other architectural structures, and subsequently introduces and investigates the prospects for autonomy of such adaptation. It does so in order to provide grounds for a detailed definition, *raison d'être* and clear direction for the development of "interactive architecture" (iA), establishing the point of departure for further research and foundation for the iA development framework.

At the outset, the broad phenomenon of architectural adaptation is examined. The given understanding of this phenomenon is based on the perspective of considering architecture to be both a product and a process of the adaptation of a human habitat to human needs. The changing nature of those needs and their bidirectional relationship with affordances of the human habitat is further discussed and contextualised (section 1.). This argument is subsequently followed by an organised overview of means, by which material adaptation of architecture is typically accomplished (section 2.). From here, the possibility of further enhancements of architecture's ability to adapt is discussed and a historical overview of concepts and projects relating to the idea of autonomous architectural adaptation is presented (section 3.). In consequence, the emergence of "interactive architecture" is discussed as a result of architectural adaptation increasing its speed and being performed autonomously (section 4.). Multiple aspects of societal relevance of interactive architecture are taken into account, based on conceptual scenarios and examples (section 5.). Key problems, risks, challenges and expectations towards interactive architecture are subsequently identified (section 6.).

The third chapter follows a postulate for an integrated approach to design, creation and operation of interactive architecture. This approach is based on considering interactive architecture to be a complex adaptive system constituted of autonomous agents, forming an actor-network of living and non-living entities. The chapter is concluded by a research strategy towards attaining such approach.

The systemic understanding of traditional architecture and of interactive architecture is firstly thoroughly scrutinised and discussed in detail on the ontological level. Following that understanding, in synchrony with examination of common architectural praxis, it is concluded that current design methods, building procedures and culture of managing buildings constrain creation of interactive architecture. These observations consequently lead to asserting the need for new strategies, methods, instruments, techniques and open building operation scenarios to be developed in conjunction with incremental specification of an integrated design[1] framework for interactive architecture. In answer to this need, a research methodology is chosen to permit constructive development of such design framework.

The chapter begins with an overview of concepts that stem from the general consideration for architecture to be a complex system made up of interrelated material objects, people and other living organisms, and immaterial (non-embodied) entities (section 1.). Successively, the concept of architectural agency (capacity of architecture to act in its environment) is analysed in context of multi-layered architectural complexity and consequent augmentation of the agency of architecture and resultant local and global adaptations (section 2.). Problems resulting from presented worldview, which are faced by architectural designers, are discussed.

[1] Kiel Moe, *Integrated Design in Contemporary Architecture*, 1st ed. (Princeton Architectural Press, 2008).

Consequently, applicability of design principles, methods and tools that are traditionally employed in architecture and other design domains to design of complex architectural adaptive systems is questioned and possible alternatives are discussed alongside latest trends in architectural design tools (section 3.). The findings of the chapter are summarised, discussed and reflected back on the research framework (section 4.).

The fourth chapter starts with a revision of the initial research framework. The chosen research approach follows the adapted grounded theory research method for building up of the iA development framework. Experimental, exploratory design case study research is employed as source of qualitative and, to a lesser extent, quantitative research data and provides means for systematic validation and adjustment of the iA framework throughout the process of its gradual formation through this and following chapters.

Consequently, the chapter primarily presents an account of a series of design research case study experiments. The chapter is driven by the initial assumption of a general and purposefully underspecified set of guidelines for creation of complex, adaptive, interactive architecture systems. These guidelines follow research presented in chapters II and III and are founded in critical evaluation of the two state-of-the-art reference projects analysed at the outset of the chapter.

As a result, throughout the selective tracings of networks of actors building up case study projects, various aspects of a process of developing interactive architecture are iteratively approached, tested and evaluated while selective focus is in turn given to key aspects of these processes. Numerous challenges and strategies, techniques and methods for addressing them are explored. Upon the termination of each project they are either rejected and replaced with new solutions or further improved and refined. This process permits an iterative build-up of a structure for a practical and efficient methodology for interactive architecture.

The chapter starts with an introduction to the method of tracing design processes and rationale behinds its choice (section 1.). Two reference state-of-the-art design cases are accordingly traced to provide a starting point and indicate biggest challenges for interactive architecture design methods (section 2.). Design case studies are subsequently discussed in five categories, corresponding to five aspects of an iA design process; experiential prototyping and realisation of designed systems (section 3.), assembling projects out of autonomous building components (section 4.), involvement of human agents in iA systems (section 5.), design of spatial organisation of complex multi-component systems (section 6.), and largely distributed projects (section 7.). In conclusion, challenges coming from these case studies are discussed, deliberating the role of a designer as working from within the iA system and showing the critical role of design instruments and experiential prototyping, which are to be further investigated in chapters V and VI.

Chapter V investigates design case study research projects from the perspective of employed design instruments and their role in the design development process. The chapter shows how design instrument prototypes have been evolving alongside the developments in design methods and how reciprocally those methods have been affected by availability or lacks in the instruments.

Various features of instruments are discussed and analysed. Initial set of function-specific instruments is consequently replaced by an ecosystem of instruments which hosts the virtual development of the designed iA system.

The chapter starts with a discussion about the role of an instrument in the architectural design process and the agency of instruments (section 1.). Following that discussion development of instruments adhering to earlier presented case studies is presented chronologically. First a number of initially dispersed endeavours are discussed (section 2.). Secondly an integrated ecosystem of protoKIT instruments is discussed (section 3.). The discussion points out the shift in approach from building multiple instruments to work on one virtual reality environment towards creating an ecosystem of non-hierarchically organised instruments. Eventually it also raises the question of the transition between virtual design and project realisation, hence providing an entry point to chapter VI.

The sixth chapter focuses on processes of project materialization. Case studies have shown high importance of deployment of working systems in early design process stages. Investigated iA system deployments included virtual and physical simulations, mock-ups, experiential prototypes, or actual realisations. In all cases early deployment proved advantageous, resulting in accelerated project development, exploration of a wider range of design solutions and delivery of more refined architectural qualities. Chapter VI investigates in more detail the problematic of extending virtual design systems into the physical world through embedded technology and rapid component fabrication.

The chapter starts with providing a generic typology of interactive building components based on earlier traced projects. Various prototyping and realisation techniques and instruments are discussed in relation to this typology (section 1.). Consequently the formation of component networks is investigated, in respect to analysing ways in which components can be interconnected in respect to physical connections as well as creation of communication channels (section 2.). Eventually the processes of gradual development of complex interactions between iA system agents across such networks are investigated (section 3.) In conclusion, the integration of design development, design instruments and prototyping instruments is discussed in correlation with challenges of interactive architecture project organisation (section 4.).

Chapter VII integrates the resulting new iA design ontology in the open and extensible “project framework” further called protoFRAME. protoFRAME is used as an ontological and organisational structure which integrates previously discussed design strategies, methods, techniques and design instruments and organises project content and planning.

Subsequently, three project templates utilising protoFRAME are proposed as starting kits for future projects. In conjunction with these templates protoFRAME lays grounds for further development of iA methods, techniques and instruments in an integrated manner, facilitating focused knowledge exchange and global evolution of iA as a distinct domain. Resulting design methods are expected to gradually build up a coherent, yet extensible and evolving methodology for complex adaptive interactive architecture. Auxiliary research directions are consequently defined in order to enable further advancement of iA design methods.

The chapter begins with a short discussion on the general concept of the project framework, its application to the conducted research and its relation to iA design methods and methodologies (section 1.). The discussion is continued by re-introducing and revising the notion of a “model” and subsequent synthesis of problems and opportunities encountered in earlier described design case study research experiments. Based on this synthesis, a set of features required from any iA system is scrutinised. From those features the core constituents and structure of the framework are derived (section 2). The details of the iA project framework “protoFRAME” are further specified, illustrated and discussed, in synchrony with earlier evaluated theories, methods and instruments (section 3). Eventually, an implementation strategy based on protoFRAME templates, involving an integration of the project framework with earlier developed iA instruments, is outlined and future challenges are discussed (section 4.)

The final, eighth chapter in a compact form re-iterates through key findings and conclusions of the dissertation and reflects them back on the research framework initially formulated in chapter I.

Bibliography

Alderfer, Clayton P. Existence, Relatedness, and Growth; Human Needs in Organizational Settings. Free Pr, 1972.

Aldrich, Frances. 'Smart Homes: Past, Present and Future'. In Inside the Smart Home, 17–39, 2003.

Alexander, Christopher. A Pattern Language: Towns, Buildings, Construction. Oxford University Press, USA, 1977.

———. 'City Is Not a Tree'. Design, 1966.

———. Notes on the Synthesis of Form. Harvard University Press, 1964.

———. The Oregon Experiment. 1ST ed. Oxford University Press, USA, 1975.

———. The Timeless Way of Building. Later printing. Oxford University Press, 1979.

Allen Orr, H. 'Turned on, a Revolution in the Field of Evolution?' The New Yorker, October 24, 2005.

Anders, Peter, and Michael Phillips. 'Arch-OS: An Operating System for Buildings'. In Proceedings of the 23rd Annual Conference of the Association for Computer Aided Design in Architecture and the 2004 Conference of the AIA Technology in Architectural Practice Knowledge Community, 282–293. Cambridge (Ontario): ACADIA, 2004.

Aranda, Benjamin, and Chris Lasch. Pamphlet Architecture 27: Tooling. 1st ed. Princeton Architectural Press, 2005.

Auyang, Sunny Y. Foundations of Complex-system Theories: In Economics, Evolutionary Biology, and Statistical Physics. Cambridge University Press, 1998.

Bachman, Leonard R. 'Architecture and the Four Encounters with Complexity'. Architectural Engineering and Design Management 4, no. 1 (March 2008): 15–30.

Backlund, Alexander. 'The Definition of System'. Kybernetes 29, no. 4 (June 1, 2000): 444–451. Banham, Reyner. The Architecture of the Well-Tempered Environment. 2nd ed. Chicago: University of Chicago Press, 1984.

Barabasi, Albert-Laszlo. Linked: How Everything Is Connected to Everything Else and What It Means. Plume, 2003.

Baran, Paul. 'Introduction to Distributed Communications Networks'. In On Distributed Communications, 1964.

Barker, Paul. 'Non-plan Revisited: Or the Real Way Cities Grow' 12, no. 2. Journal of Design History (1999).

Batty, Michael. Cities and Complexity: Understanding Cities with Cellular Automata, Agent-Based Models, and Fractals. The MIT Press, 2005.

Beesley, Philip, and Rachel Armstrong. 'Soil and Protoplasm: The Hylozoic Ground Project'. Architectural Design 81, no. 2 (March 1, 2011): 78–89.

Beesley, Philip, and Omar Khan. Situated Technologies Pamphlets 4: Responsive Architecture, Performing Instruments. New York: The Architectural League of New York, 2009.

Bennett, Jane. Vibrant Matter: A Political Ecology of Things. Duke University Press Books, 2009.

Von Bertalanffy, Ludwig. General System Theory: Foundations, Development, Applications. Revised. George Braziller, 1976.

———. 'The Theory of Open Systems in Physics and Biology'. Science 111, no. 2872 (1950): 23–29.

Betsky, Aaron, Ole Bouman, David Deutsch, Elizabeth Diller/Ricardo Scofidio, Monika Fleischmann/Wolfgang Strauss, Sulan Kolatan/William Mac Donald, William J. Mitchell, et al. Disappearing Architecture: From Real to Virtual to Quantum. 1st ed. Birkhäuser Basel, 2005.

Bier, Henriette. ‘System-Embedded Intelligence in Architecture’. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia, 292–303. Heijningen: Jap Sam Books, 2012.

Bier, Henriette, Kathleen de Bodt, and Jerry Galle. ‘SC: Prototypes for Interactive Architecture’. In Proceedings of the 12th International Conference on Interactive Technologies and Sociotechnical Systems, 21–28. VSMM’06. Berlin, Heidelberg: Springer-Verlag, 2006.

Biloria, Nimish. Adaptive Corporate Environments: Creating Real-time Interactive Spatial Systems for Corporate Offices Incorporating Computation Techniques. Delft: TU Delft, Faculty of Architecture, 2007.

———. ‘InfoMatters, a Multi-agent Systems Approach for Generating Performative Architectural Formations’. International Journal of Architectural Computing 9, no. 3 (September 1, 2011): 205–222.

———. ‘Interactive Morphologies: An Investigation into Integrated Nodal Networks and Embedded Computation Processes for Developing Real-time Responsive Spatial Systems’. Frontiers of Architectural Research 1, no. 3 (September 2012): 259–271.

Biloria, Nimish, and Kas Oosterhuis. ‘Envisioning the Responsive Milieu: An Investigation into Aspects of Ambient Intelligence, Human Machine Symbiosis and Ubiquitous Computing for Developing a Generic Real-time Interactive Spatial Prototype’. In CAADRIA 2005, Proceedings of the 10th International Conference on Computer Aided Architectural Design Research in Asia. New Delhi, 2005.

Bittermann, Michael S., I. Sevil Sariyildiz, and Özer Ciftcioglu. ‘A Computational Intelligence Approach to Alleviate Complexity Issues in Design’. In Complexity Theories of Cities Have Come of Age, edited by Juval Portugali, Han Meyer, Egbert Stolk, and Ekim Tan, 347–368. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

Blundell-Jones, Peter, Doina Petrescu, and Jeremy Till. Architecture and Participation. Taylor & Francis, 2005.

Bongers, Bert. ‘Multimodal Interaction in Protospace 1.0’. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia, 533–548. Heijningen: Jap Sam Books, 2012.

Bratman, Michael E. Intention, Plans, and Practical Reason. Center for the Study of Language and Inf, 1999.

Brodey, Warren. ‘The Design of Intelligent Environments: Soft Architecture’. Landscape 17, no. 1 (1967).

Brownell, Blaine. Transmaterial: A Catalog of Materials That Redefine Our Physical Environment. 1st ed. Princeton Architectural Press, 2005.

Bullivant, Lucy. 4dsocial: Interactive Design Environments. Wiley, 2007.

———. 4dspace: Interactive Architecture. 1st ed. Academy Press, 2005.

———. Responsive Environments: Architecture, Art and Design. Victoria and Albert Museum, 2006.

Callaghan, Vic, Graham Clarke, and Jeannette Chin. ‘Some Socio-technical Aspects of Intelligent Buildings and Pervasive Computing Research’. Intelligent Buildings International 1 (2009): 56–74.

Carroll, Sean B. Endless Forms Most Beautiful: The New Science of Evo Devo and the Making of the Animal Kingdom. 1st ed. W. W. Norton & Company, 2005.

Caspersen, C J, M A Pereira, and K M Curran. 'Changes in Physical Activity Patterns in the United States, by Sex and Cross-sectional Age'. Medicine and Science in Sports and Exercise 32, no. 9 (September 2000): 1601–1609.

Cayci, Filiz, Victor Callaghan, and Graham Clarke. 'A Distributed Intelligent Building Agent Language (DIBAL)'. Orlando, Florida, 2000.

Charmaz, Kathy. Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis. 1st ed. Sage Publications Ltd, 2006.

Clarke, Roger. 'Asimov's Laws of Robotics: Implications for Information Technology-Part I'. Computer 26 (1993): 53–61.

———. 'Asimov's Laws of Robotics: Implications for Information Technology-Part II'. Computer 27 (1994): 57.

Coen, Michael. 'Design Principles for Intelligent Environments'. In Proceedings of the Fifteenth National Conference on Artificial Intelligence, 547–554. Madison, WI, 1998.

Cohen da Lara, Max, and Hans Hubers. 'Muscle Body'. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia, 409–415. Heijningen: Jap Sam Books, 2012.

Conrads, Ulrich. Programs and Manifestoes on 20th-Century Architecture. The MIT Press, 1975.

Cook, Diane, and Sajal Das. Smart Environments: Technology, Protocols and Applications. Wiley-Interscience, 2004.

Coradeschi, S., and A. Saffiotti. 'Symbiotic Robotic Systems: Humans, Robots, and Smart Environments'. Intelligent Systems, IEEE 21, no. 3 (2006): 82–84.

Creswell, John W. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. 3rd ed. Sage Publications, Inc, 2008.

Crompton, Dennis. Concerning Archigram. New edition. Archigram Archives, 1999.

Cross, Nigel. Engineering Design Methods: Strategies for Product Design. 4th ed. Wiley, 2008.

———. The Automated Architect. Methuen, 1980.

———. 'The method in their madness; Understanding how designers think'. Delft, the Netherlands, 1996.

Dawkins, Richard. The Selfish Gene. 2nd ed. Oxford University Press, USA, 1990.

Deleuze, Gilles. Thousand Plateaus. Continuum International Publishing Group, 2004.

Derix, Christian, Åsmund Gamlesæter, Pablo Miranda, Lucy Helme, and Karl Kropf. 'Simulation Heuristics for Urban Design'. In Digital Urban Modeling and Simulation, edited by Stefan Müller Arisona, Gideon Aschwanden, Jan Halatsch, and Peter Wonka, 242:159–180. Communications in Computer and Information Science. Springer Berlin Heidelberg, 2012.

Dow, S., B. MacIntyre, J. Lee, C. Oezbek, J.D. Bolter, and M. Gandy. 'Wizard of Oz Support Throughout an Iterative Design Process'. IEEE Pervasive Computing 4, no. 4 (December 2005): 18 – 26.

Downes, Edward J., and Sally J. McMillian. 'Defining Interactivity: A Qualitative Identification of Key Dimensions'. New Media Society 2, no. 2 (June 1, 2000): 157–179.

Eastman, Charles. 'Adaptive-conditional Architecture'. In Design Participation, Proceedings of the Design Research Society's Conference, 51–57. Manchester: Academy Editions, 1972.

Eastman, Chuck, Paul Teicholz, Rafael Sacks, and Kathleen Liston. BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors. Wiley, 2008.

Eng, K., A. Babler, U. Bernardet, M. Blanchard, M. Costa, T. Delbruck, R.J. Douglas, et al. 'Ada - Intelligent Space: An Artificial Creature for the SwissExpo.02'. In IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA '03, 3:4154 – 4159 vol.3, 2003.

Feng, Han. 'Quantum Architecture'. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia, 341–359. Heijningen: Jap Sam Books, 2012.

Fenn, Jackie, and Mark Raskino. Mastering the Hype Cycle: How to Choose the Right Innovation at the Right Time. Harvard Business School Press, 2008.

Ferber, Paul, Franz Foltz, and Rudy Pugliese. 'Interactivity Versus Interaction: What Really Matters for State Legislature Web Sites?' Bulletin of Science Technology Society 25, no. 5 (October 1, 2005): 402–411.

Fisk, Nicholas. Trillions. New edition. Hodder Children's Books, 1998.

Fitchett, Anne. 'Function and Creativity in Architecture'. Urban Forum 9, no. 2 (June 1998): 271–278.

Foque, Richard. Building Knowledge in Architecture. ASP - Academic & Scientific Publishers, 2010.

Fox, Michael, and Miles Kemp. Interactive Architecture. Princeton Architectural Press, 2009.

Frazer, John. An Evolutionary Architecture. Architectural Association Publications, 1995.

———. 'The Architectural Relevance of Cybernetics'. Systems Research 10, no. 3 (1993): 43–48.

Friedenthal, Sanford, Alan Moore, and Rick Steiner. A Practical Guide to SysML: The Systems Modeling Language. Revised. Morgan Kaufmann, 2009.

Friedman, Yona. 'Information Processes for Participatory Design'. In Design Participation. Proceedings of the Design Research Society's Conference. Manchester: Academy Editions, 1972.

———. Yona Friedman: Pro Domo. Actar, 2006.

Friedrich, Christian. 'Immediate Architecture'. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia, 225–260. Heijningen: Jap Sam Books, 2012.

Friedrich, Christian, and Owen Slootweg. 'Protospace 4.0'. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia. Heijningen: Jap Sam Books, 2012.

Fuller, Matthew, and Usman Haque. Situated Technologies Pamphlets 2: Urban Versioning System 1.0. New York: The Architectural League of New York, 2008.

Georgeff, Michael, Barney Pell, Martha Pollack, Milind Tambe, and Michael Wooldridge. 'The Belief-Desire-Intention Model of Agency'. In Intelligent Agents V: Agents Theories, Architectures, and Languages, 1–10, 1999.

Gershenfeld, N, and D. Cohen. 'Internet 0: Interdevice Internetworking - End-to-End Modulation for Embedded Networks'. Circuits and Devices Magazine, IEEE 22, no. 5 (2006): 48–55.

Giedion, Siegfried. Space, Time and Architecture: The Growth of a New Tradition, Fifth Revised and Enlarged Edition. 5th ed. Harvard University Press, 1967.

Gijsbers, Roel. Aanpasbaarheid Van De Draagstructuur : Veranderbaarheid Van De Drager Op Basis Van Gebruikerseisen in Het Kader Van Slimbouwen. Eindhoven: Bouwstenen Publicatiebureau, 2011.

Gilbert, Nigel. Agent-Based Models. annotated edition. Sage Publications, Inc, 2007.

Goldman, John A. 'Towards a Systems Biology of the Built Environment'. Energy and Buildings 2, no. 4 (December 1979): 309–321.

Goldstein, S.C., J.D. Campbell, and T.C. Mowry. 'Programmable Matter'. Computer 38, no. 6 (May 2005): 99 – 101.

Habraken, John. 'The Control of Complexity' 4, no. 2. Representations and Realities (July 1, 1987): 3–15.

Hagras, Hani, Victor Callaghan, and Martin Colley. 'Intelligent Embedded Agents'. Information Sciences 171, no. 4 (May 13, 2005): 289–292.

Hanington, Bruce, and Bella Martin. Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions. Rockport Publishers, 2012.

Haque, Usman. 'Architecture, Interaction, Systems'. Arquitetura & Urbanismo, 2006.

Herd, Kate, Andrew Bardill, Mehmet Karamanoglu, Kate Herd, Andrew Bardill, and Mehmet Karamanoglu. 'Development of a Design Probe to Reveal Customer Touch Points in the Sale of Mass Customised Products.' Design Principles and Practices, 2009.

Heylighen, Francis. 'The Science Of Self-Organization And Adaptivity'. Organizational Intelligence and Learning, and Complexity, in: The Encyclopedia of Life Support Systems (EOLSS) (1999): 253–280.

Holland, John. Hidden Order: How Adaptation Builds Complexity. First Edition. Basic Books, 1996.

Hosale, MarkDavid. 'Protospace 3.0'. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia, 557–560. Heijningen: Jap Sam Books, 2012.

Hosale, MarkDavid, and Chris Kievid. 'An InteractiveWall as a Prototype for an Emotive Architectural Component'. In : IA#3, edited by Oosterhuis Kas, Owen Slootweg, and Xin Xia. Heijningen: Jap Sam Books, 2010.

Hughes, Jonathan, and Simon Sadler. Non-Plan: Essays on Freedom, Participation and Change in Modern Architecture and Urbanism. Architectural Press, 2000.

Huisbergem, E.D., and I. Kriens. 'Plan Cycle'. In The Essence of Spatial Planning and Strategy. TU Delft, 2003.

Iwamoto, Lisa. Digital Fabrications: Architectural and Material Techniques. 144 p. Princeton Architectural Press, 2009.

Jablonski, Andrzej, Ryszard Klempous, and Benedykt Licznerski. 'Diversified Approach to Methodology and Technology in Distributed Intelligent Building Systems'. In Computer Aided Systems Theory - EUROCAST 2003, 174–184, 2003.

Jacobs, Jane. The Death and Life of Great American Cities. Modern Library, 1993.

Jaskiewicz, T., A. van der Helm, and W. Aprile. 'Creative Approach to the Design and Prototyping of Experimental Smart Spaces, Case Studies from the Interactive Environments Minor'. In Smart Spaces and Next Generation Wired/Wireless Networking. Vol. 6294. Lecture Notes in Computer Science. St. Petersburg, Russia: Springer, 2010.

Jaskiewicz, T., and N. Schueler. 'Prototype Presentation: Interactive Portals Hyperbody'. In The Architectural Annual 2007-2008, Delft University of Technology, edited by L. Calabrese, A. Doeschate, A. Geenen, D. Hauptmann, J. Heintz, and U. Knaack, 48–51. Delft, 2009.

Jaskiewicz, Tomasz. '(In:)forming Interactive Architectural Systems, Case of the xMAiA Meta-model'. In ACADIA 10: LIFE In:formation, On Responsive Information and Variations in Architecture. New York: ACADIA, 2010.

———. 'Architecture as a Multi-Agent System'. In Volume#28, Internet of Things, 56–60. Archis, 2011.

———. 'Behavioral Modeling Applied to the Speed and Friction: Automotive Complex Project'. In iA#3 - Interactive Architecture Emotive Styling, edited by Kas Oosterhuis and Xia Xia. Rotterdam: Jap Sam Books, 2010.

———. 'Complex Multiplayer Urban Design System – Concept and Case Studies'. Edited by M. Forsell, M. Alexander, H. X. Lin, A. Knüpfer, R. Prodan, L. Sousa, and A. Streit. Euro-Par 2009 – Parallel Processing Workshops 6043/2010. Lecture Notes in Computer Science (2009).

———. 'Dynamic Design Matter[s]: Practical Considerations for Interactive Architecture'. Harvard University Graduate School of Design, Cambridge (USA), 2008.

———. '"iPortals" as a Case Study Pre-Prototype of an Evolving Network of Interactive Spatial Components'. In Proceedings of the 28th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA). Minneapolis, 2008.

———. 'Paracity – A Digital Urban Design Process'. Delft, the Netherlands: Episode Publishers, 2006.

Jaskiewicz, Tomasz, and Christian Friedrich. 'Muscle Space'. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia, 420–424. Heijningen: Jap Sam Books, 2012.

Jaskiewicz, Tomasz, and Kas Oosterhuis, eds. 751 - Multiplayer Design Studio. China Architecture & Building Press, 2008.

Jaskiewicz, Tomasz, and Kas Oosterhuis. '798 Multiplayer Design Game, A New Tool for Parametric Design'. In Space Time Play: Computer Games, Architecture and Urbanism: The Next Level, edited by F. von Borries, S. P. Walz, M Böttger, D. Davidson, H. Kelley, and J. Kücklich. 1st ed. Birkhäuser Architecture, 2007.

Jencks, Charles, and Karl Kropf. Theories and Manifestoes of Contemporary Architecture. 2nd ed. Academy Press, 2006.

Jensen, J. F. 'Interactivity. Tracking a New Concept. In Media and Communication Studies'. Nordicom Review 1/1998. Special Issue: The XIII Nordic Conference on Mass Communication Research 1, no. 19 (1998).

Johnson, Philip C. Writings. 1st Edition. Oxford University Press, 1979.

Johnson, Steven. Emergence: The Connected Lives of Ants, Brains, Cities, and Software. Scribner, 2002.

Jones, John Chris. Design Methods. 2nd ed. Wiley, 1992.

Kahn, Louis. Louis Kahn: Essential Texts. W. W. Norton & Company, 2003.

Kelley, Tom. 'Prototyping Is the Shorthand of Innovation' 12, no. 3. Design Management Journal (2001): 35–42.

Kelly, Kevin. Out of Control: The New Biology of Machines, Social Systems, & the Economic World. Basic Books, 1995.

Kenrick, Douglas T., Vladas Griskevicius, Steven L. Neuberg, and Mark Schaller. 'Renovating the Pyramid of Needs'. Perspectives on Psychological Science 5, no. 3 (May 1, 2010): 292 –314.

Kieran, Stephen, James Timberlake, Stephen Kieran, and James Timberlake. Refabricating Architecture: How Manufacturing Methodologies Are Poised to Transform Building Construction. 1st ed. McGraw-Hill Professional, 2003.

Kievid, Chris. 'Protospace 4.0 Pavilion'. Zeppelin 92 (2011).

Kievid, Chris, and Christian Friedrich. 'Digital Pavilion, Intelligent Architecture'. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia. Heijningen: Jap Sam Books, 2012.

Kievid, Chris, and Kas Oosterhuis. 'Muscle NSA, a Basis for a True Paradigm Shift in Architecture'. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia. Heijningen: Jap Sam Books, 2012.

Kievid, Chris, Dieter Vandoren, Kas Oosterhuis, Han Feng, and Xin Xia. 'iLITE, a Creative Vision on Architectural Lighting Solutions'. In iA#4 : Quantum Architecture. Heijningen: Jap Sam Books, 2011.

Kilian, Axel. Design Exploration Through Bidirectional Modeling of Constraints. MIT Press, 2006.

Kilian, Axel, and John Frazer. 'Particle-spring Systems for Structural Form Finding'. Journal of International Association for Shell and Spatial Structures 46, no. 147 (2005).

Kline, David. 'The Embedded Internet'. Wired, October 1996.

Knabb, Ken. Situationist International Anthology. Revised & Expanded. Bureau Of Public Secrets, 2007.

Kolarevic, Branko. 'Exploring Architecture of Change', 2009.

Kolarevic, Brankop. 'Towards Integrative Design'. International Journal of Architectural Computing 7, no. 3 (September 1, 2009): 335–344.

Koolhaas, Rem. Delirious New York: A Retroactive Manifesto for Manhattan. The Monacelli Press, 1997.

Korebrits, Sander. 'Muscle Bamboostic'. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia, 416–419. Heijningen: Jap Sam Books, 2012.

Koseki, Michihiko, Kengo Minami, and Norio Inou. 'Cellular Robots Forming a Mechanical Structure'. In Distributed Autonomous Robotic Systems 6, 139–148, 2007.

Kraal, Ben J. 'Actor-network Inspired Design Research: Methodology And Reflections' Conference Paper, 2007.

Van Kranenburg, Rob. Network Notebook #2: The Internet of Things, 2007.

Kretzer, Manuel, and Dino Rossi. 'ShapeShift'. Leonardo (2012): 480–481.

Krier, Leon. The Architecture of Community. 1st ed. Island Press, 2009.

Kronenburg, Robert. Flexible: Architecture That Responds to Change. Laurence King Publishers, 2007.

Krumm, John. Ubiquitous Computing Fundamentals. Chapman and Hall/CRC, 2009.

Kuhn, Thomas S. The Structure of Scientific Revolutions. 1st ed. University Of Chicago Press, 1996.

Kuniavsky, Mike. Observing the User Experience: A Practitioner’s Guide to User Research. 1st ed. Morgan Kaufmann, 2003.

Kuznetsov, Stacey, Eric Paulos, and Mark D Gross. ‘WallBots : Interactive Wall-Crawling Robots In the Hands of Public Artists and Political Activists’. Human-Computer Interaction (2010): 208–217.

Kwinter, Sanford. Architectures of Time: Toward a Theory of the Event in Modernist Culture. The MIT Press, 2002.

De Landa, Manuel. A Thousand Years of Nonlinear History. Zone Books, 2000.

Latour, Bruno. Reassembling the Social: An Introduction to Actor-Network-Theory. Oxford University Press, USA, 2007.

Latour, Bruno, and Albena Yaneva. ‘Give Me a Gun and I Will Make All Buildings Move: An ANT’s View of Architecture’. In Explorations in Architecture: Teaching, Design, Research, 80–89. Basel: Birkhäuser, 2008.

Laurel, Brenda. The Art of Human-Computer Interface Design. Addison-Wesley Professional, 1990.

Lawson, Bryan. How Designers Think, Fourth Edition: The Design Process Demystified. 4th ed. Architectural Press, 2005.

Lefebvre, Henri. The Production of Space. Wiley-Blackwell, 1992.

Lelieveld, Charlotte, and Liek Voorbij. ‘Dynamic Material Application for Architectural Purposes’. Advances in Science and Technology 56 (2008): 595–600.

Lénárd, Ilona. ‘Powerlines’. In iA#2. Episode Publishers, Rotterdam, 2009.

Leonard, John. Systems Engineering Fundamentals. DIANE Publishing, 1999.

Lidwell, William, Kritina Holden, and Jill Butler. Universal Principles of Design. Rockport Publishers, 2003.

Luhmann, Niklas. Social Systems. 1st ed. Stanford University Press, 1996.

Lynch, Kevin. The Image of the City. The MIT Press, 1960.

Lynn, Greg. Animate Form. 1st ed. Princeton Architectural Press, 1999.

Manovich, Lev. The Language of New Media. The MIT Press, 2002.

Markussen, Thomas, and Thomas Birch. ‘Minding Houses’, n.d. Accessed April 18, 2010.

Maslov, A.H. ‘A Theory of Human Motivation’. Psychological Review 50 (1943): 370–396.

Mathews, Stanley. From Agit-prop to Free Space: The Architecture of Cedric Price. 1st ed. Black Dog Publishing, 2007.

Maxwell, Joseph A. Qualitative Research Design: An Interactive Approach. 2nd ed. Sage Publications, Inc, 2004.

McCullough, Malcolm. Digital Ground: Architecture, Pervasive Computing, and Environmental Knowing. New Ed. The MIT Press, 2005.

McLuhan, Marshall. Understanding Media. 2nd ed. Routledge, 2001.

Meadows, Mark Stephen. Pause & Effect: The Art of Interactive Narrative. New Riders Press, 2002.

Merz, Hermann, Thomas Hansemann, and Christof Hübner. ‘Introduction to Building Automation’. In Building Automation, 1–25, 2009.

Miles, Matthew B., and A. Michael Huberman. Qualitative Data Analysis: An Expanded Sourceboo. 2nd ed. Sage Publications, Inc, 1994.

Miller, John H., and Scott E. Page. Complex Adaptive Systems: An Introduction to Computational Models of Social Life. illustrated edition. Princeton University Press, 2007.

Mitchell, William J. City of Bits: Space, Place, and the Infobahn. The MIT Press, 1996.

Moe, Kiel. Integrated Design in Contemporary Architecture. 1st ed. Princeton Architectural Press, 2008.

Moffatt, Sebastian, and Niklaus Kohler. 'Conceptualizing the Built Environment as a Social–ecological System'. Building Research & Information 36, no. 3 (2008): 248.

Moggridge, Bill. Designing Interactions. 1st ed. The MIT Press, 2007.

Mondada, Francesco, Giovanni C. Pettinaro, Andre Guignard, Ivo W. Kwee, Dario Floreano, Jean-Louis Deneubourg, Stefano Nolfi, Luca Maria Gambardella, and Marco Dorigo. 'Swarm-Bot: A New Distributed Robotic Concept'. Autonomous Robots 17, no. 2 (2004): 193–221.

Mozer, M. C. 'An Intelligent Environment Must Be Adaptive'. IEEE Intelligent Systems and Their Applications 14, no. 2 (1999).

———. 'Lessons from an Adaptive House'. In Smart Environments: Technologies, Protocols, and Applications, edited by D. Cook and Das. Hoboken, NJ: J. Wiley & Sons, 2005.

Muffatto, Moreno. Open Source: A Multidisciplinary Approach. Imperial College Press, 2006.

Negroponte, Nicholas. Soft Architecture Machines. The MIT Press, 1976.

———. The Architecture Machine: Toward a More Human Environment. The MIT Press, 1973.

Neufert, Ernst, Peter Neufert, Bousmaha Baiche, and Nicholas Walliman. Architects' Data. 3rd ed. Wiley-Blackwell, 2002.

Neumayr, Robert R. 'Topotransegrity. Non-Linear Responsive Environments'. In Game Set And Match II. On Computer Games, Advanced Geometries, and Digital Technologies, edited by Kas Oosterhuis and Lukas Feireiss, 427–437. Episode Publishers, 2006.

Nieuwenhuys, Constant. 'New Babylon'. Haags Gemeetenmuseum, 1974.

Norman, Donald A. 'Design Thinking: A Useful Myth - Core77', 2010. http://www.core77.com/blog/columns/design_thinking_a_useful_myth_16790.asp.

———. Living with Complexity. The MIT Press, 2010.

Novak, Marcos. 'Liquid Architectures in Cyberspace'. In Cyberspace: First Steps, 225–254. MIT Press, 1991.

———. 'Transvergence: Finite and Infinite Minds'. In Game Set and Match II: The Architecture Co-laboratory on Computer Games, Advanced Geometries, and Digital Technologies, edited by Kas Oosterhuis and Lukas Feireiss. Episode Publishers Rotterdam, 2006.

Novak, Marcos, and Alessandro Ludovico. 'Interview with Markos Novak', 2001.

Oosterhuis, Kas. Architecture Goes Wild: Manifest Writings. 010 Uitgeverij, 2001.

———. Hyperbodies, Towards an E-motive Architecture. Edited by Antonino Saggio. 1st ed. Birkhäuser Basel, 2003.

———. New Kind of Building Non-standard Arch: A Designers Guide for Non-Standard Architecture. NAi Publishers, 2011.

———. ONLogic: Speed and Vision The Master Architect Series. Images Publishing Group Pty. Ltd., 2008.

———. 'Protospace 1.0'. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia, 516–532. Heijningen: Jap Sam Books, 2012.

———. Towards a New Kind of Building: A Designers Guide for Non-Standard Architecture. NAi Publishers, 2011.

Oosterhuis, Kas, and Nimish Biloria. 'Interactions with Proactive Architectural Spaces: The Muscle Projects'. Commun. ACM 51, no. 6 (June 2008): 70–78.

Oosterhuis, Kas, Christian Friedrich, Tomasz Jaskiewicz, and Dieter Vandoren. 'Protospace Software'. Delft, the Netherlands: TU Delft, 2007.

Oosterhuis, Kas, Chris Kievid, Christian Friedrich, and Tomasz Jaskiewicz. 'Protospace 2.0'. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia, 549–556. Heijningen: Jap Sam Books, 2012.

Oosterhuis, Kas, and Xin Xia. iA #1. episode publishers, 2007.

Ophir, Yaniv. 'Collective Intelligence: An Agent-Based Approach to Programmatic Organization in Architecture'. International Journal of Architectural Computing 7, no. 3 (2009): 479–499.

Ottchen, Cynthia. 'The Future of Information Modelling and the End of Theory: Less Is Limited, More Is Different'. Architectural Design 79, no. 2 (2009): 22–27. doi:10.1002/ad.845.

Pask, Gordon. Conversation Theory, Applications in Education and Epistemology, 1976.

———. Conversation Theory, Applications in Education and Epistemology, 1976.

———. 'The Architectural Relevance of Cybernetics'. Architectural Design (1969).

Pastore, Marco, and Valentina Sabatelli. Leisurator TM, 2004.

Patrick, Healy. The Model and Its Architecture. 010 Uitgeverij, 2008.

Pickering, Andrew. 'Material Culture and the Dance of Agency'. In The Oxford Handbook of Material Culture Studies. Oxford University Press, 2010.

Piker, Daniel. 'Kangaroo Physics | Grasshopper'. Accessed July 20, 2012. http://www.food4rhino.com/project/kangaroo.

Portugali, J. Complex Artificial Environments: Simulation, Cognition and VR in the Study and Planning of Cities. 1st ed. Springer, 2006.

Prigogine, Ilya, and G. Nicolis. Self-Organization in Nonequilibrium Systems: From Dissipative Structures to Order Through Fluctuations. First Edition. John Wiley & Sons, 1977.

Rabeneck, Andrew. 'Cybermation: A Useful Dream'. Architectural Design (1969).

Rafaeli, Sheizaf. 'Interactivity: From New Media to Communication'. In Advancing Communication Science: Merging Mass and Interpersonal Processes, edited by R. P. Hawkins, J. M. Wiemann, and S. Pingree, 16:110–134. Sage Annual Review of Communication Research. Sage Pubns, 1988.

Rafaeli, Sheizaf, and Fay Sudweeks. 'Networked Interactivity'. Journal of Computer-Mediated Communication 4, no. 4 (1997).

Reiser, Jesse. Atlas of Novel Tectonics. 1st ed. Princeton Architectural Press, 2006.

Riehle, Dirk. Framework Design: A Role Modeling Approach. ETH Zurich, 1999.

Roberts, Don, and Ralph Johnson. 'Evolving Frameworks: A Pattern Language for Developing Object-Oriented Frameworks'. In Proceedings of the Third Conference on Pattern Languages and Programming. Addison-Wesley, 1996.

Ronzani, Daniel. 'The Battle of Concepts: Ubiquitous Computing, Pervasive Computing and Ambient Intelligence in Mass Media'. Ubiquitous Computing and Communication Journal 4, no. 3 (2009).

Rosenblueth, Arturo, Norbert Wiener, and Julian Bigelow. 'Behavior, Purpose and Teleology'. Philosophy of Science 10, no. 1 (1943): 18–24.

Russell, Stuart J., and Peter Norvig. Artificial Intelligence: A Modern Approach. Prentice Hall, 2009.

Saggio, Antonino. 'Interactivity at the Centre of Avant-Garde Architectural Research'. Architectural Design 75, no. 1 (2005): 23–29.

———. The IT Revolution in Architecture, Thoughts on a Paradigm Shift. Lulu Press, 2010.

———. 'The Search for an Information Space'. edited by Kas Oosterhuis and Lukas Feireiss, 214–219. Delft, the Netherlands: Episode Publishers, Rotterdam, 2006.

Sanders, Elizabeth B.-N., and Pieter Jan Stappers. 'Co-creation and the New Landscapes of Design'. CoDesign 4, no. 1 (2008): 5–18.

Schuler, Douglas, and Aki Namioka. 'Contextual Inquiry: A Participatory Technique for System Design'. In Participatory Design: Principles and Practices. Routledge, 1993.

Schumacher, Patrik S. The Autopoiesis of Architecture: A New Framework for Architecture. 1st ed. Wiley, 2011.

Senagala, Mahesh. 'Building Is a Network for Living in: Toward New Architectures'. In Proceedings of the 2005 Annual Conference of the Association for Computer Aided Design In Architecture, 36–47. Savannah (Georgia), 2005.

Shan, Yin. Applications of Complex Adaptive Systems. 1st ed. IGI Publishing, 2008.

Shannon, Claude E., and Warren Weaver. The Mathematical Theory of Communication. University of Illinois Press, 1971.

Shepard, Mark, ed. Sentient City: Ubiquitous Computing, Architecture, and the Future of Urban Space. New York: The Architectural League of New York, 2011.

Shoham, Yoav, and Kevin Leyton-Brown. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press, 2008.

Slootweg, Owen. 'Muscle Tower II'. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia, 404–408. Heijningen: Jap Sam Books, 2012.

Sobek, Werner, and Patrick Teuffel. 'Adaptive Systems in Architecture and Structural Engineering' (July 30, 2001): 36–45.

Speed, Chris. 'An Internet of Things That Do Not Exist'. Interactions 18, no. 3 (May 2011): 18–21.

Sutherland, Ivan. 'Sketchpad: A Man-machine Graphical Communication System'. Technical Report no. 574 (1963 2003).

Teuffel, Patrick. 'From Adaptive to High-performance Structures'. Seoul, Korea, 2011.

Upton, Eben, and Gareth Halfacree. Raspberry Pi User Guide. John Wiley & Sons, 2012.

Vastenburg, M.H., D.V. Keyson, and H. de Ridder. 'Interrupting People at Home'. In 2004 IEEE International Conference on Systems, Man and Cybernetics, 1:59 – 64 vol.1, 2004.

Venturi, Robert. Robert Venturi: Complexity and Contradiction in Architecture. 2nd ed. The Museum of Modern Art, New York, 1977.

Venturi, Robert, Steven Izenour, and Denise Scott Brown. Learning from Las Vegas - Revised Edition: The Forgotten Symbolism of Architectural Form. Revised. The MIT Press, 1977.

Verde, Marco. 'Performative Design Processes for Architecture [P]a'. In Hyperbody: First Decade of Interactive Architecture, edited by Kas Oosterhuis, Henriette Bier, Nimish Biloria, Chris Kievid, Owen Slootweg, and Xin Xia, 304–328. Heijningen: Jap Sam Books, 2012.

Verde, Marco, MarkDavid Hosale, Jelle Feringa, Ruairi Glynn, and Bob Sheil. 'Investigations in Design and Fabrication at Hyperbody'. In Fabricate: Making Digital Architecture, 98–105. London: Riverside Architectural Press, 2011.

Vidler, Anthony. 'Toward a Theory of the Architectural Program'. October - (October 1, 2003): 59–74.

Vitruvius. Vitruvius: Ten Books on Architecture. Cambridge University Press, 2001.

Wagner, Ferdinand, Ruedi Schmuki, Thomas Wagner, and Peter Wolstenholme. Modeling Software with Finite State Machines: A Practical Approach. Auerbach Publications, 2006.

Wahba, Mahmoud A., and Lawrence G. Bridwell. 'Maslow Reconsidered: A Review of Research on the Need Hierarchy Theory'. Organizational Behavior and Human Performance 15, no. 2 (April 1976): 212–240.

Waldrop, M. Mitchell. Complexity: The Emerging Science at the Edge of Order and Chaos. 1st ed. Simon & Schuster, 1992.

Weaver, W. A Quarter Century in the Natural Sciences. The Rockefeller Foundation Annual Report. The Rockefeller Foundation, 1958.

Weilkiens, Tim. Systems Engineering with SysML/UML: Modeling, Analysis, Design. Morgan Kaufmann, 2008.

Weiser, Mark, R. Gold, and J.S. Brown. 'The Origins of Ubiquitous Computing Research at PARC in the Late 1980s'. IBM Systems Journal 38, no. 4 (1999).

Wiener, Norbert. Cybernetics, Second Edition: Or the Control and Communication in the Animal and the Machine. 2nd ed. The MIT Press, 1965.

Wigley, Mark. Constant's New Babylon: The Hyper-architecture of Desire. 010 Publishers, 1998.

Wooldridge, M., and W. van der Hoek. 'Multi-Agent Systems'. In Handbook of Knowledge Representation, edited by Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, 887–928. Elsevier Science, 2008.

Wooldridge, Michael. An Introduction to MultiAgent Systems. 2nd ed. Wiley, 2009.

Yang, Ang. Intelligent Complex Adaptive Systems. 1st ed. IGI Publishing, 2008.

Zegher, Catherine de, and Mark Wigley. The Activist Drawing: Retracing Situationist Architectures from Constant's New Babylon to Beyond. illustrated edition. The MIT Press, 2001.

Zuk, William and Roger H Clark. Kinetic Architecture. Von Nostrand Reinhold Company., 1970.

Appendix 1 – project credits

751 City

Studio: Hyperbody MSc3 / TU Delft Bouwkunde
Teachers: prof. Kas Oosterhuis, Tomasz Jaskiewicz, Nimish Biloria, Gerrie Hobbelman, Bigge Tuncer, Guus Westgeest
Remote teachers: prof. Dong Wei, prof. Xu Weiguo, Zhang Qian, Du Rong
Students: Da Da Wang, Christina Rangel, Marije van der Laag, Nora Schuler, Miroslava Tumova, Roman Krajger, Christian Mack, Tomas Lapka, Alexander Baumann, Jedrek Kolensinski, Ewoud Ruifrok, Jenna Fizel, Eva Kiesel, Felix Wurst, Isil Sencar, Magdalena Garbarczyk, Sema Alagam Aslan, Feng Lin, Julien Westerfeld, Lukas Mahlknecht, Astrid Mitterlehner, Tobias Schimme, Theresa Schütz
Remote students: Zang Gongxiu, Wang Ke, Jing Jing, Zhang Zijun, Liu Di, Chen Xiaoji, Liang Qiwei, Tao Xiaochen, Xiong Xiong, Zhou Shi
Date: 2006, quarter 3-4

Bubble Pods Portal

Studio: Interactive Technology Design MSc2 / TU Delft Industrial Design Engineering | ID-StudioLab
Teachers: Walter Aprile, Aadjan van der Helm
Students: Thomas E. Louts, Tjeerd J. IJtsma, Wei Zhang
Date: 2008, quarter 1-2

Cockpit

Design: ONL [Oosterhuis_Lénárd]
Lead designe: Kas Oosterhuis
Project architect: Sander Boer
Design team: Kas Oosterhuis, Ilona Lénárd, Cas Aalbers, Sander Boer, Tom Hals, Ines Moreira, Dimitar Karanikolov, Vladin Petrov, Tom Smith, Richard Lewis, Gijs Joosen, Andrei Badescu, Maciek Swiatkowsky
Client: Hessing BV
Date: 2005
Site: Utrecht Leidsche Rijn

Curtain Portal

Studio: Vertical studio Hyperbody BSc6+MSc3 / TU Delft, Bouwkunde
Teachers: prof. Kas Oosterhuis, Tomasz Jaskiewicz, Dieter Vandoren, Nora Schueler, Jan Willem Breider
Students: Robyn Bruins, Marcus Chaidez, Daiki Nakagawa, Maria Vera van Embden Andres, Erwin van Osch
Date: 2008, quarter 1-2

Distributed Faculty

Studio: Hyperbody MSc3 / TU Delft Bouwkunde
Teachers: prof. Kas Oosterhuis, Tomasz Jaskiewicz, Nora Schueler
Students: Efe Gozen, Sander Korebrits, Wulf Kramer, Gustavo Nascimento, Querien Velter, Bao An Nguyen Phuoc, Aurelie Hsiao, Seol Mingyu, Jonas Po Sing Sin, Junjie Yan, Tade Godbersen, Krzysztof Gornicki, Agata Kycia, Erwin Osch, Roxana Palfi, Catarina Ramel, Martha Maatkamp, Harikrishnan Sasidharan, Gabriela Semeco, Jiayu Tjong, Wai Wing Yun
Date 2008, quarter 3-4

Festo Headquarters

Design: ONL [Oosterhuis_Lénárd]
Lead design: Kas Oosterhuis, Ilona Lénárd
Project architect: Gjis Joosen
Design team: Tomasz Jaskiewicz, Bernard Sommer, Owen Slootweg, Martin Bering Henriksen
Site: Zollberg Sud, Germany
Date: 2008

GEN

Studio: Interactive Environments Minor / TU Delft Bouwkunde|Hyperbody, Industrial Design Engineering|ID-StudioLab
Teachers: Tomasz Jaskiewicz (BK), Aadjan van der Helm (IDE), Walter Aprile (IDE), prof. Kas Oosterhuis (BK), MarkDavid Hosale (BK), Dieter Vandoren(BK), Rob Luxen (IDE)
Guest teachers: refunc, Jerome Decock, Daan Roosegaarde, Ruairi Glynn, Tetsuo Tomiyama
Students: Altynay Imanbekova, Joris Hoogeboom, Ben van Wijk, Hanneke Hoogewerf, Tino de Bruijn, Patrick Pijnappel, Alexander de Mulder
Date: 2010, quarter 3-4

protoKIT

Concept: Tomasz Jaskiewicz
Execution: multiple authors, open contributions
Date: 2011

iWEB

Design: ONL [Oosterhuis_Lénárd]
Design team: Kas Oosterhuis, Ilona Lénárd, Sander Boer, Pim Marsman, Dieter Vandoren, Gerard van den Engel, Ronald Brandsma, Marthijn Pool
Client: TU Delft, Faculty of Architecture
Date: 2006
Site: Delft, TU Delft, Faculty of Architecture

Jealous Portal

Studio: Interactive Technology Design MSc2 / TU Delft, Industrial Design Engineering | ID-StudioLab
Teachers: Walter Aprile, Aadjan van der Helm
Students: Daniël Hagmeijer, Venty Vergianti, Alexandra Vosmaer
Date: 2008, quarter 1-2

Leaves Portal

Studio: Vertical studio Hyperbody BSc6+MSc3 / TU Delft Bouwkunde
Teachers: prof. Kas Oosterhuis, Tomasz Jaskiewicz, Dieter Vandoren, Nora Schueler, Jan Willem Breider
Students: Giacomo Destefanis, Gosha Jacewicz, Michiel Susebeek, Tade Godbersen, Taejong Jeong
Date: 2008, quarter 1-2

Manhal Oasis

Design: ONL [Oosterhuis_Lénárd]
Project architect: Prof ir. Kas Oosterhuis
Design team: Kas Oosterhuis, Ilona Lénárd, Tomasz Jaskiewicz, Gijs Joosen, Rafael Seemann,

Michael Gorzynscki, Barbara Janssen, Rena Logara, Lidia Badarnah, Jan Gasparik, Tomasz Sachanowicz, Satish Kumar, Dimitar Karanikolov
General consultants: Anita Mehra Homayoun, Amir Mehra
Engineering consultant: ARUP
Hotel design consultant: Mahmoudiehconcepts
Client: International Capital Trading LLC
Site: Manhal Palace Abu Dhabi, UAE
Date: 2006

Muscle Trans-Ports

Design: ONL [Oosterhuis_Lénárd], Hyperbody
Lead design: Prof ir Kas Oosterhuis
Design team: Kas Oosterhuis, Ilona Lénárd, Bert Bongers, Chris Kievid, Laura Aquili, Remko Siemerink, Sven Blokker
Engineering: ONL, d3bn, festo, buitink
Client: Biennale 2000 Venice, mnam/cci Centre Pompidou Paris
Date: 2003
Site: Centre Pompidou Paris

Muscle Façade

Studio: Hyperbody BSc5 / TU Delft Bouwkunde
Teachers: Tomasz Jaskiewicz, prof. Kas Oosterhuis
Students: Sebastian Baggelaar, Michael Bolier, Po-Chun Huang, Harm Sollie, Axel vanZalingen
Date: 2007, quarters 1-2

Muscle Space

Studio: Hyperbody BSc5 / TU Delft Bouwkunde
Teachers: Christian Friedrich, Tomasz Jaskiewicz, prof. Kas Oosterhuis
Students: Maarten Feberwee, Sander Janssen, Arjan Klem, Youval Kuipers, René de Rooij, Wouter Streefkerk Edwin Uytenbroek
Date: 2007, quarters 3-4

Odyssey

Studio: Interactive Environments Minor / TU Delft Bouwkunde|Hyperbody, Industrial Design Engineering|ID-StudioLab
Teachers: Tomasz Jaskiewicz (BK), Aadjan van der Helm (IDE), Walter Aprile (IDE), prof. Kas Oosterhuis (BK), MarkDavid Hosale (BK), Dieter Vandoren(BK), Rob Luxen (IDE)
Guest teachers: refunc, Jerome Decock, Daan Roosegaarde, Ruairi Glynn, Tetsuo Tomiyama
Students: Govert Flint, Lieke Kraan, Bob Groeneveld, Jesse Timmermans, Merijn Pen, Thomas van Oekelen, Melisa Garza Vales
Date: 2010, quarter 3-4

Paracity

Studio: Hyperbody MSc4 / TU Delft Bouwkunde
Supervisor: prof. Ir. Kas Oosterhuis
Co-supervisors: Stephen Read, Alexander Vollebregt
Author: Tomasz Jaskiewicz
Date: 2005, quarter 1-2

protoCOLOGY

Studio: Hyperbody vertical studio BSc6, MSc2 / TU Delft Bouwkunde
Teachers: Christian Friedrich, Chris Kievid
Students: Rene-Paul van Leeuwen, Michel Stienstra, sander Apperlo, Gerben Knol, Igor Leffertstra, Marjolein Overtoom, Jasper Schaap, Jaimy Siebel, Wilson Wong, Frank van Brunschot, Bao Nguyen Phuoc
Date: 2009 quarter 3-4

protoDECK (1.0)

Conceptual design: prof. Kas Oosterhuis, Chris Kievid, Marco Verde, MarkDavid Hosale
Parametric form and fabrication design: Marco Verde
Electronic systems, sensors, light implementation: MarkDavid Hosale
Date: 2010

protoDECK 2.0

Conceptual redesign: Tomasz Jaskiewicz, Stefan Dulman, Andrei Pruteanu
students (EWI): Agostino di Figlia, Harm Jan Treep, Chiel de Roest, Steffan Karger, Sjors van Berkel
Electronics engineering: Rob Luxen
Microcontroller sponsor: NXP
Assistance: Veronika Laszlo, Gary Chang, Sina Mostafavi, Mariana Popescu
Date: 2012

protoFAB

Coordinators (chronologically): Marco Verde, Christian Friedrich, Chris Kievid
Date: 2010 onwards

protoMAP

Concept: Tomasz Jaskiewicz
Implementation: Veronika Laszlo, Tomasz Jaskiewicz, Christian Friedrich
Date: 2011 onwards

protoSPACE 1.0 (demo)

Team: prof. Kas Oosterhuis, Hans Hubers, Sven Blokker, Misja van Veen, Chris Kievid
Date: 2003

protoSPACE 1.1 (demo)

Team: prof. Kas Oosterhuis, Hans Hubers, Tomasz Jaskiewicz, Dieter Vandoren
Date: 2004

protoSPACE 1.2 (demo)

Team: prof. Kas Oosterhuis, Bert Bongers, Tomasz Jaskiewicz, Dieter Vandoren, Christian Friedrich, Yolande Harris
Date: 2004

protoSPACE 1.3 (demo)

Team: prof. Kas Oosterhuis, Bert Bongers, Tomasz Jaskiewicz, Dieter Vandoren, Christian Friedrich, Yolande Harris
Date: 2005

protoSPACE 1.4 (demo)

Team: prof. Kas Oosterhuis, Bert Bongers, Tomasz Jaskiewicz, Dieter Vandoren, Christian Friedrich, Yolande Harris
Date: 2005

protoSPACE 3.0 laboratory

Coordinators (chronologically): MarkDavid Hosale, Tomasz Jaskiewicz
Date: 2010 onwards

protoSPACE 4.0 pavilion

Head architect: prof. Kas Oosterhuis
Semester coordinator: Chris Kievid
Lead teacher: Christian Friedrich
Design teacher: Gijs Joosen
Studio: Hyperbody MSc2 / TU Delft Bouwkunde
Students: Yang Shi, Viss Naoum, Jonas Sin, Jun Jie Yan, Roxana Palfi, Urvi Sheth, Kristof Gornicki, Soran Park, Stella Lam, Mingyu Seol, Melina Mezari, Kwok-Tung Chun, Agata Kycia, Aurélie Hsiao, Gustavo Nascimento, Erwin Osch, Marco Cimenti, Harikrishnan Sasidharan
Scripting expert: Owen Slootweg
Structural expert: Bas Wijnbeld
Interaction expert: Mark David Hosale
Fabrication expert: Marco Verde
Material expert: Charlotte Lelieveld
Project team fabrication: Chris Kievid, Owen Slootweg, Jelle Feringa
Date: 2009 onwards

protoTAG

Concept: Christian Friedrich
Implementation: Christian Friedrich, Veronika Laszlo, Tomasz Jaskiewicz
Date: 2010 onwards

protoWIKI

Concept: Tomasz Jaskiewicz
Implementation: Veronika Laszlo, Tomasz Jaskiewicz, Christian Friedrich
Date: 2011 onwards

reNDSM

Studio: Hyperbody MSc1 / TU Delft Bouwkunde
Teachers: prof. Kas Oosterhuis, Tomasz Jaskiewicz, Chris Kievid, Florian Eckhardt
Students: Wen Tao Bi (Max), Eric Geboers, Vahid Ghodsi, Sjors de Graaf, Gergely Hory, Dustin Huang, Miriam Polak, Akshay Rajan, Harish Ramakrishnan, Pim Schachtschabel, Lotte Suijker, Linus Tan, Romain Thijsen, Sam van Til, Teun Verkerk, Chao Wang, Amid Parsi, Magdalena Melon, Dezhang Zhou
Website: http://rendsm.hyperbody.nl
Date: 2012, quarter 1-2

Salzburg National Park Centre / Schmetterling Wingman

Design: ONL [Oosterhuis_Lénárd]
Design team: Kas Oosterhuis, Ilona Lénárd, Sander Boer, Gijs Joosen, Pim Marsman, Cas Aalbers, Alessandro de Santis, Sanne Plomp

Client: Land Salzburg (competition)
Date: 2005
Site: Mittersill, Austria

sCAPE, LEDworks

Studio: Interactive Environments Minor / TU Delft Bouwkunde|Hyperbody, Industrial Design Engineering|ID-StudioLab
Teachers: Tomasz Jaskiewicz (BK), Aadjan van der Helm (IDE), Walter Aprile (IDE), prof. Kas Oosterhuis (BK), MarkDavid Hosale (BK), Dieter Vandoren(BK), Rob Luxen (IDE)
Guest teachers: refunc, Jerome Decock, Daan Roosegaarde, Ruairi Glynn, Tetsuo Tomiyama
Students: Iris van Loon, Marieke Dijksma, Teun Verkerk, Fons van der Berg, Tom Goijer, Cees-Willem Hofstede
Website: http://ledwork.org/
Date: 2010, quarter 3-4

Sound Barrier

Design: ONL [Oosterhuis_Lénárd]
Lead design: Kas Oosterhuis, Ilona Lénárd
Project architect: Cas Aalbers
Design team: Cas Aalbers, Sander Boer, Tom Hals, Dimitar Karanikolov, Tom Smith, Richard Lewis, Barbara Janssen, Gijs Joosen, Andrei Badescu, Maciek Swiatkowski, Rafael Seemann
Client: Projectbureau Leidsche Rijn Utrecht
Production: Meijers Staalbouw bv
Date: 2006
Site: Utrecht Leidsche Rijn

Speed and Fricition Automotive Complex

Design: ONL [Oosterhuis_Lénárd]
Project architect: Prof ir Kas Oosterhuis
Design team: Kas Oosterhuis, Ilona Lénárd, Gijs Joosen, Cas Aalbers, Sander Boer, Tomasz Jaskiewicz, Chris Kievid, Dieter Vandoren, Barbara Janssen, Henrike Michler, Eirini Logara, Han Feng, Brenda Vonk Noordegraaf
Client: confidential
Site: Abu Dhabi
Date: 2005

Urban Body – Building Relations

Studio: Urban Body MSc2 / TU Delft Bouwkunde
Teachers: Tomasz Jaskiewicz, Nimish Biloria
Students: Marco Boeber, Gijs Braakman, Ivonne Weichold, Julia Rubin, Nora Schueler, Rannveig Yli, Stella Dourtme, Tomohito Naito
Date: 2006, quarter 1-2

VHpark

Studio: Hyperbody MSc1 / TU Delft Bouwkunde
Teachers: Tomasz Jaskiewicz, Christian Friedrich, prof. Kas Oosterhuis
Students: Koen Kegel, Joost Noorden, Niklas Ruprechter, Jacob Lam Chak, Lieke Kraan, Liviu Teodorescu, Ioli Plastira, Felipe Aldana, Jan Top, Miyushi van Hijfte, Alvin Järving, Alvin, Zhu Wei, Katja Virta, René Brakels, Manon Tardieu, Manuel Zucchi, André Dessens, Frederich Steenkamp, Michael Zhe Zhang, Fani Ntintoka, Jeannette Bisseling, Anurag Bhattacharya, Esther Odijk, Wilton Li, Mohammed Al-Khalili, Kelwin Palmer, Merwin deBruin, Anna

Marcassoli, Mirjam Wiechers, Nate Weems, Nathaniel, Lubomir Peytchev
Website: http://vhpark.hyperbody.nl
Date: 2011, quarter 3-4

Appendix 2 – figure and image credits

Fig. 13. Courtesy to Christopher Alexander / Oxford University Press

Fig. 14. Courtesy to Rachel Whiteread

Fig. 16. Courtesy to building Smart Alliance, source:
http://buildingsmart-tech.org/ifc/IFC2x4/rc2/html/index.htm

Fig. 17. Courtesy to Tim Weilkiens / Morgan Kaufman press

Fig. 18. Courtesy to Bill Moggridge / the MIT Press

Fig. 24.-28. Courtesy to ONL[Oosterhuis_Lénárd]

Fig. 46 Authored by Rannveig Yli

Fig. 48 Authored by Roman Krajger

Fig. 49b. Authored by Alexander Baum

Fig. 50a. Authored by Agata Kycia

Fig. 50b. Authored by Jonas Sin Po Sing

Fig. 50c. Authored by Bao Ang Nguyen Phouc

Fig. 51. Authored by Efe Gozen, Sander Korebrits, Wulf Kramer, Gustavo Nascimento, Querien Velter, Bao An Nguyen Phuoc, Aurelie Hsiao, Seol Mingyu, Jonas Po Sing Sin, Junjie Yan, Tade Godbersen, Krzysztof Gornicki, Agata Kycia, Erwin Osch, Roxana Palfi, Catarina Ramel, Martha Maatkamp, Harikrishnan Sasidharan, Gabriela Semeco, Jiayu Tjong, Wai Wing Yun

Fig. 55. Courtesy to Bert Bongers

Fig. 59. Courtesy to Paul Baran / RAND

Img. 3a. Courtesy to/by Yona Friedman

Img. 3b. Courtesy to the Canadian Centre for Architecture / by Cedric Price

Img. 3c. Courtesy to/by Peter Cook

Img. 4a. Guy Debord "Psychogeographic guide to Paris" cover, public domain

Img. 4b. Courtesy to Constant Nieuwenhuijs/010 Publishers

Img. 5. Courtesy to ONL[Oosterhuis_Lénárd]

Img. 7a. Courtesy to Walter Aprile

Img. 7b. Courtesy to Aldo Hoeben

Img. 12. Authored by André Dessens, Frederich Steenkamp, Michael Zhe Zhang

Img. 13. Authored by Niklas Ruprechter, Jacob Lam Chak, Lieke Kraan, Liviu Teodorescu

Img. 15. Courtesy to ONL[Oosterhuis_Lénárd]

Img. 17b. Authored by Harish Ramakrishnan

Img. 17c. Authored by Linus Huang

Img. 18. Courtesy to Bert Bongers

Img. 19., 21. Courtesy to Hyperbody

Img. 23b. Courtesy to Stefan Dulman

Other figures and images were created by the author.

Appendix 3 – referenced websites

'3DVIA Virtools - Dassault Systèmes', http://www.3ds.com/products/3dvia/3dvia-virtools/.

'Architecture :: Space and Mass -- Britannica Online Encyclopedia', http://www.britannica.com/EBchecked/topic/32876/architecture/31847/Space-and-mass.

'Arduino', http://www.arduino.cc/.

'Effector'. The Free Dictionary, http://www.thefreedictionary.com/effector.

'Fluidic Muscle DMSP/MAS'. 2012. http://xdki.festo.com/xdki/data/doc_ENGB/PDF/EN/DMSP-MAS_EN.PDF.

'Geco | Grasshopper'. 2012. http://www.grasshopper3d.com/group/geco.

'IFC Overview Summary', http://www.buildingsmart-tech.org/specifications/ifc-overview/ifc-overview-summary.

'Instructables - Make, How To, and DIY', http://www.instructables.com/.

'Internet of Things - An Action Plan for Europe, Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions'. Commission of the European Communities, 2009. http://ec.europa.eu/information_society/policy/rfid/documents/commiot2009.pdf.

Jan Korbes, and Denis Oudendijk. 'Refunc', http://www.refunc.nl.

Janssen, Theo. 'Strandbeest', http://www.strandbeest.com/.

'Materia', http://materia.nl/.

'Material ConneXion', http://www.materialconnexion.com/.

Mau, Bruce. 'An Incomplete Manifesto for Growth', 1998. http://www.brucemaudesign.com/manifesto.html.

'Mecanoo Website ("mekel Park, Campus Tu Delft" Project)', http://www.mecanoo.nl.

'Phidgets Inc.', http://www.phidgets.com/.

'Phidgets Inc. - Unique and Easy to Use USB Interfaces', http://www.phidgets.com/.

Preisinger, Clemens. 'Karamba | Grasshopper', http://www.food4rhino.com/project/karamba.

Piker, Daniel. 'Kangaroo Physics | Grasshopper', http://www.food4rhino.com/project/kangaroo.

'Project - Definition and More from the Free Merriam-Webster Dictionary', http://www.merriam-webster.com/dictionary/project.

'Relation - Definition and More from the Free Merriam-Webster Dictionary', http://www.merriam-webster.com/dictionary/relation?show=0&t=1309181372.

Somlai-Fsicher, Adam, and Usman Haque. 'Reconfigurable House', http://house.propositions.org.uk/.

Swoyer, Chris. 'Properties'. In The Stanford Encyclopedia of Philosophy, edited by Edward N. Zalta. 2009. http://plato.stanford.edu/archives/win2009/entries/properties/.