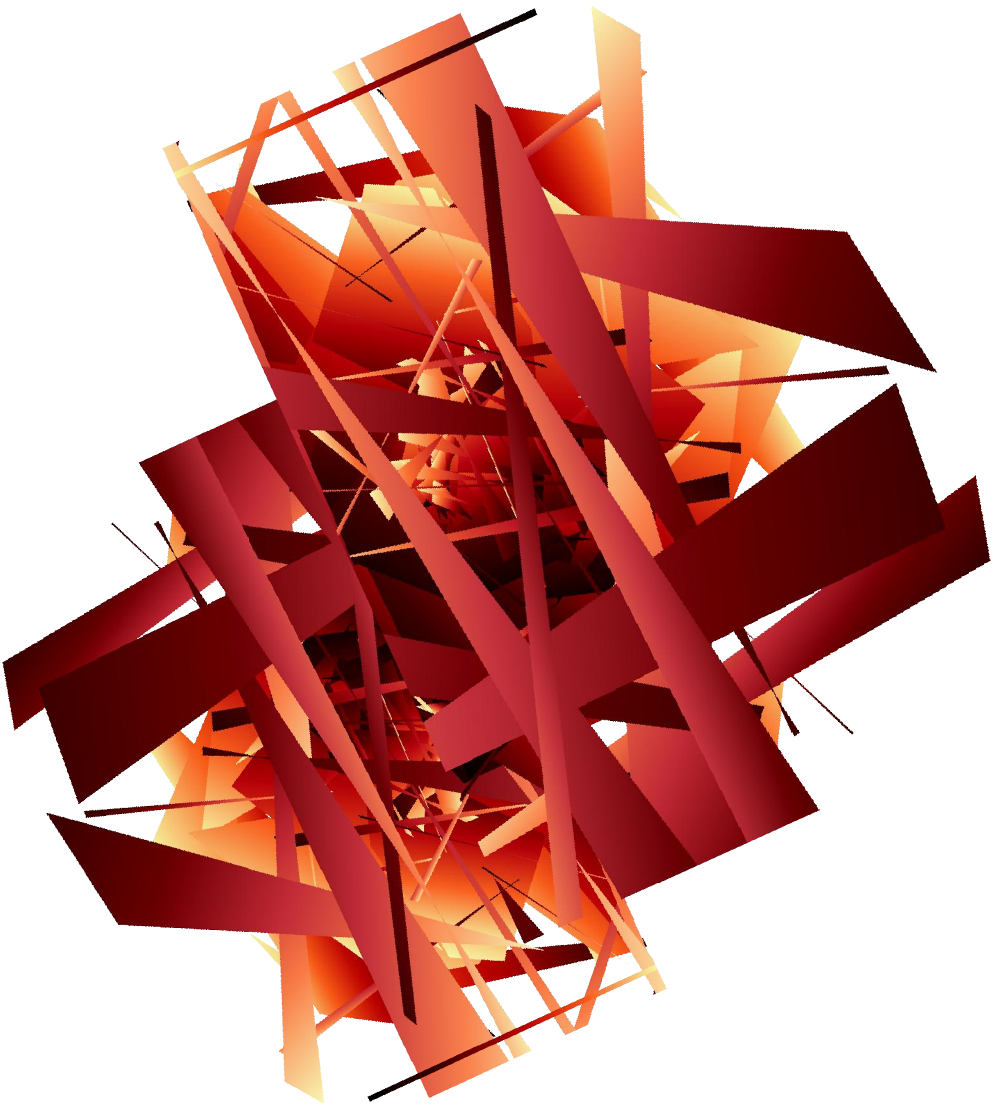
Evolutionary, Unconscious Design Support

for the Architectural, Engineering and Construction Industry

Martinus van de Ruitenbeek



Evolutionary, Unconscious Design Support

Evolutionary, Unconscious Design Support for the Architectural, Engineering and Construction industry

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 dinsdag 4 september 2012 om 10:00 uur
door Hendrik Klaas Martinus VAN DE RUITENBEEK
civiel ingenieur
geboren te Johannesburg, Zuid-Afrika.

Dit proefschrift is goedgekeurd door de promotor:
Prof. H.A.J. de Ridder

Copromotor: Dr. M.R. Beheshti

Samenstelling promotiecommissie:

Rector Magnificus,	Voorzitter
Prof.dr.ir. H.A.J. de Ridder,	Technische Universiteit Delft, promotor
Dr.ir. M.R. Beheshti,	Technische Universiteit Delft, copromotor
Prof.dr. I. Horváth,	Technische Universiteit Delft
Prof.dr.ir. A.C.J.M. Eekhout,	Technische Universiteit Delft
Prof. C.M.F. Dirven,	Erasmus Medisch Centrum
Dr. E. Dado,	Nederlandse Defensie Academie
Dr. J. Donovan,	Dublin Institute of Technology
Prof.drs.ir. J.K. Vrijling,	Technische Universiteit Delft

De volgende bedrijven hebben een belangrijke bijdrage geleverd door een combinatie van werken en promoveren mogelijk te maken:
Volker InfraDesign (Woerden)
Spie Controlec Engineering (Terneuzen)

ISBN 978-90-808007-0-0

Acknowledgements

"Lay a firm foundation with the bricks that others throw at you." - David Brinkley

My loving Mirjam and my exceptional children Cassandra and Nathaly ceaselessly supported me during the journey, for which I am very thankful. Their warmth and dynamics were important stimuli up to the very end. I am also grateful to my parents who went to great lengths in creating the conditions that enabled me to achieve this dream.

During the entire process, Reza Beheshti was much more than a supervisor. He guided my dreams, ideas and illusions as a professional scientific artist, but also became a true friend. Thank you for the wonderful gift of discovering how small that piece of dust is that I attempted to stack onto the pile of scientific knowledge !

With his vivid mind, Edwin Dado helped me to re-think many topics that affected both this thesis and my personal views considerably. He is a pleasant and inspiring friend who lives life with a passionate laugh.

Raymond Turner kindly initiated a visit to Dublin and established a fruitful cooperation with John Donovan. As a molecular biologist, John took a giant leap into the world of the AEC industry. He acquiescently explained many of the nuances of nature's adaptivity and extensively advised on how to implement them in the context of this thesis.

Design is a very complex field of science in which Prof. Horváth is an authority. His deep insights and nuanced views were as very helpful in completing this thesis. I have learnt a great deal from his dedicated attitude towards science and his modest open-heartedness in our discussions.

Although neurosurgery is a very different from AEC design, Prof. Dirven empathised himself in the subject and proposed to study the effects of diseases in designers' products. This useful addition has so many aspects that it may serve as an interesting cross-disciplinary subject for future PhD research projects.

Prof. Eekhout has an impressive reputation in the art of true one-of-a-kind product development, both as a scientist and as a professional. Being in that exceptional position, he provided me with useful insights about balancing at the edge of innovation and reuse within a competitive environment.

In various sessions with master students, I came to know prof. Vrijling as a designer pur sang who deeply believes in the heuristic power of structural designers. His many anecdotes helped me in attempts to prevent useful models from becoming extinct.

It was important for me to combine research with working. Fortunately, Ron Brökling was so kind to enable this during the first year. Later, Sipke Huitema supported me in many ways. He is a notorious optimist with an astonishing amount of detailed knowledge behind a most tranquil character.

Finally, it was an honour to have been a PhD student under the supervision of the passionate prof. Hennes de Ridder. Our conversations always covered a plethora of subjects in which he left much room for alternative and conflicting views. His belief in the subject and his humour were great incentives.

I would like to warmly thank the many unmentioned colleagues, friends, students, advisers, family and the FrameDesign users who supported me either directly or indirectly. Without the help of so many people it would have been infeasible to compile this booklet.

Summary

> Being is enthralling ! Personified, cultivated social communities imbue fiction, silently diffuse reality and whisk by science. Internees unconsciously cogitate in vanquishing epitomes and syndicate in leviathan evolution.

The Architecture, Engineering and Construction (AEC) industry is a complex system in which carpenters, structural designers, architects, modellers, cost estimators, planners, politicians and many others act apart together in project-specific virtual enterprises. There is a large amount of actors, an overwhelming number of ongoing processes, distributed, decentralised organisations and a variety of projects. This complicates efficient communication and supply chain integration which, according to yearly estimates, leads to a waste of between 10% and 50%.

Inspired by mass production industries such as the car and ship industries, researchers introduced techniques such as Systems Engineering (SE), Building Information Modelling[1] (BIM), product modelling and object libraries in the AEC industry. It is being expected that these techniques will significantly reduce the waste if all actors in the supply chain conform to them. However, the past 20 years have shown that attempts to impose such unifying techniques on the AEC system only partially rendered the expected results. Perhaps the premises that the AEC industry is comparable to

[1] Copied from Product Lifecycle Management (PLM)

a mass production industry and that it is possible to create widely accepted standards that enable full chain integration, should be revisited.

An alternative premise is that the AEC industry has the characteristics of a Complex Adaptable System (CAS). Under the pressure of design code changes, new contract forms, economical crises, new technologies, competition and such, it bends, adapts, survives and continues to produce infrastructure and buildings. Nature and the Internet are comparable examples of CAS. For example, the Internet hosts many actors, there are many processes going on, it is distributed, decentralised and it hosts a rich variety of applications / standards. Despite of major distortions[2] and waste due to redundant storage, spam, hacking and such, this chaotic system remains fast and reliable, and continues to produce a wealth of information.

An important feature of the Internet is that it allows users to search for information, copy it and respond to it (e.g. with ratings and comments). Structural designers often lack such features for design-related information within their own enterprises. Therefore a relevant addition is to allow structural designers for lavishly and effortlessly finding, copying, publishing, comparing and improving historical products in order to mobilise enterprise knowledge and to allow for products to evolve.

If products are copied often and if they are rated well, then they rise in ranking[3] and become eligible to be certified. The certified products are the pieces of information that people are capable of managing and overseeing: the enterprise Conscious. Finding solutions to new problems (i.e. finding products, either from historical cases or from certified products) is the art of designers' Unconscious. Within this socio-evolutionary system, ambient intelligence forms an active mind and products evolve.

Introducing a new socio-evolutionary standard at this point would violate the CAS principle that new imposed standards are unlikely to remain unchanged. Enterprises rather should create competing socio-evolutionary systems together with humanoid interfaces. The systems that best succeed at maximising a) product evolution and b) providing design assistance[4],

---

[2] For example, when Egypt forced providers to cut off the internet, new or alternative services like Speak to Tweet, Tor and SailMail quickly appeared.

[3] Much like apps in a smartphone market.

[4] e.g. by comparing new problems with historical problems and their solutions, or by communicating in natural language

will survive. Thus, both products and their host environments are socio-evolutionary systems.

In order to gather evidence for the above theory, various prototypes were designed, of which a simple finite element analysis tool "FrameDesign" (Section 3.1) was the most noteworthy. With this tool, structural designers can create, calculate and share frame structures in the cloud, after which fellow structural designers are able to copy and improve them. The tool keeps track of this product evolution and visualises it to the structural designers. To date, there were ~50k active users (~160k total downloads), and ~23% of the structures they shared were evolving.

The prototypes demonstrate only a small portion of the theory, and it is currently impossible to relate their use to a reduced waste rate in the AEC industry. However, future researchers are encouraged to create similar frameworks and to test them on a larger scale. Perhaps, such frameworks that allow for socio-evolving products will lead to reduced waste due to

- efficient use of the corporate mind (case based reasoning)
- emergence of more and better certified products that lead to more repetition (evolution)
- use of existing standards (humanoid communication)

In a future scenario, with having a statistically sufficient amount of (enterprise-specific) evolving products, the win-win socio-evolutionary system (Leviathan) may provide design support in the form of human-like heuristics. Sufficiently rich heuristic design support systems, combined with natural interfaces (such as voice and vision interaction) for effortless communication with humans, may form one or more useful virtual design assistants. These assistants may conglomerate in virtual enterprises (which is common to the AEC industry). This is a field of science that is known as artificial social intelligence.

Samenvatting

Het zijn is intrigerend ! Gepersonifieerde, gecultiveerde sociale gemeenschappen infiltreren fictie, maken de werkelijkheid ongemerkt diffuus en streven wetenschap voorbij. Ingewijden overpeinzen onbewust in overwinnende epitomen en verenigen in Leviathaanse evolutie.

Evolutionaire, Onderbewuste Ontwerpondersteuning voor de Bouw

De bouw industrie is een complex systeem waarin timmerlieden, ontwerpers, architecten, vormgevers, calculators, planners, politici en vele anderen korte samenwerkingsverbanden aangaan in project-specifieke virtuele ondernemingen. Er is een grote hoeveelheid actoren, een overweldigend aantal lopende processen, gedistribueerde, gedecentraliseerde organisaties en een verscheidenheid aan projecten. Dit bemoeilijkt efficiënte communicatie en ketenintegratie, wat volgens jaarlijkse schattingen leidt tot faalkosten van tussen 10% en 50%.

Geïnspireerd door massaproductie industrieën zoals de auto en scheepvaart industrie, introduceerden onderzoekers technieken zoals Systems Engineering (SE), Building Information Modelling[5] (BIM), productmodellering en objectenbibliotheken in de bouw. De verwachting is dat deze technie-

[5] Gekopieerd van Product Lifecycle Management (PLM)

ken de faalkosten aanzienlijk zullen verminderen als alle actoren in de keten zich hieraan conformeren. Echter de afgelopen 20 jaar hebben laten zien dat dergelijke pogingen tot uniformering in de bouw slechts gedeeltelijk het beoogde resultaat opleveren. Mogelijk moet het uitgangspunt dat de bouw te vergelijken is met een massa-industrie waarin uniformering in de gehele keten gerealiseerd kan worden, herzien worden.

Een alternatief uitgangspunt is dat de bouw de kenmerken van een Complex Aanpasbaar Systeem (CAS) heeft. Onder druk van veranderende regelgeving, nieuwe contractvormen, economische crises, nieuwe technologieën, concurrentie en dergelijke, buigt het, past het zich aan, overleeft het en blijft het infrastructuur en gebouwen produceren. De natuur en het internet zijn vergelijkbare voorbeelden van CAS. Op het internet zijn er bijvoorbeeld vele actoren aanwezig, zijn er veel processen gaande, is er een gedecentraliseerde organisatie en een grote verscheidenheid aan toepassingen en standaarden. Ondanks grote verstoringen[6] en faalkosten als gevolg van redundante opslag, spam, hack pogingen en dergelijke, blijft dit chaotische systeem op een snelle en betrouwbaar manier een schat aan informatie leveren.

Een belangrijk kenmerk van het internet is dat gebruikers informatie kunnen zoeken, kopiëren en (bijvoorbeeld met beoordelingen en commentaar) kunnen reageren. Constructieve ontwerpers beschikken vaak niet over vergelijkbare mogelijkheden voor bedrijfsspecifieke producten, en daarom is het een relevante aanvulling om het mogelijk te maken dat zij een overvloed aan relevante historische producten moeiteloos kunnen vinden, kopiëren, publiceren, vergelijken en verbeteren. Op deze manier wordt bedrijfskennis gemobiliseerd en kunnen producten gaan evolueren.

Als producten worden vaak gekopieerd worden en voorzien worden van een goede beoordeling, dan zullen ze stijgen in populariteit[7] en in aanmerking komen voor certificering. De gecertificeerde producten zijn de stukjes informatie die mensen nog kunnen overzien: het Bewustzijn van een bedrijf. Het vinden van oplossingen voor nieuwe problemen (dwz het vinden van producten, historisch of gecertificeerd) is de kunst van het Onderbewuste van ontwerpers. Binnen dit sociaal-evolutionaire systeem vormt contextgevoelige intelligentie een actieve geest en evolueren producten.

---

[6] Toen bijvoorbeeld Egypte internet providers forceerde om het internet af te sluiten, ontstonden er snel alternatieven zoals Speak to Tweet, Tor and SailMail.

[7] Vergelijkbaar met apps in een smartphone market.

Het introduceren van een nieuw sociaal-evolutionaire standaard is weinig zinvol omdat standaarden in CAS zelden ongewijzigd te blijven. In plaats daarvan zouden bedrijven concurrerende sociaal-evolutionaire systemen met mensachtige interfaces moeten maken. De systemen die het best slagen in het maximaliseren van a) product evolutie en b) het bieden van ontwerpondersteuning[8] zullen overleven. Daarom zijn zowel producten als de sociaal-evolutionaire omgevingen aan evolutie onderhevig.

Om bewijs voor de bovenstaande theorie te verzamelen zijn diverse prototypes ontworpen, waaronder een eenvoudige eindige elementen analyse tool "FrameDesign" (Sectie 3.1). Met deze tool kunnen constructieve ontwerpers eenvoudige 2D structuren maken en delen, waarna collega-ontwerpers deze kunnen kopiëren en verbeteren. De tool houdt de product-evolutie bij en visualiseert dit aan de ontwerpers. Tot op heden waren er ~ 50k actieve gebruikers (~ 160K downloads in totaal), en ~ 23% van de structuren die ze deelden waren aan evolutie onderhevig.

De prototypes tonen slechts een klein deel van de theorie aan, en het is in dit stadium onmogelijk om het gebruik ervan te relateren aan de faalkosten in de bouw. Toekomstige onderzoekers worden daarom aangespoord om soortgelijke theorieën te creëren en te testen op grotere schaal. Wellicht kullen dergelijke sociaal-evolutionaire systemen leiden tot minder faalkosten door

- efficiënt gebruik van het bedrijfscollectief (case based reasoning)
- ontstaan van meer en betere gecertificeerde producten die leiden tot meer herhaling (evolutie)
- gebruik van de bestaande standaarden (menselijke communicatie)

In een toekomstscenario waarin er een voldoende grote hoeveelheid (bedrijfsspecifieke) evoluerende producten aanwezig zijn voor statistiek, kan de win-win sociaal-evolutionair systeem (Leviathan) mogelijk ontwerpondersteuning in de vorm van mens-achtige heuristiek bieden. Voldoende rijke heuristische ontwerpsystemen, gecombineerd met natuurlijke interfaces (zoals spraak en visuele interactie) voor een moeiteloze communicatie met mensen, kunnen mogelijk een of meer nuttige virtuele ontwerp-assistenten

[8] bijv door nieuwe problemen met bestaande te vergelijken of door via natuurlijke communicatie te interacteren.

vormen en eigen virtuele ondernemingen (wat heel gewoon is in de bouw) samenstellen. Dit is het wetenschappelijke terrein van kunstmatige sociale intelligentie.

List of abbreviations

AEC	Architecture, Engineering and Construction
BIM	Building Information Model
CAD	Computer Aided Design
DNA	Deoxyribonucleic Acid (carrier of genetic information)
DSS	Design Support System(s)
CAS	Complex Adaptable System(s)
CBR	Case Based Reasoning
DSS	Design Support System
FEM	Finite Element Method
GUI	Graphical User Interface
HMI	Human-Machine Interface (or MMI, Man-Machine Interface)
IATS	Information At The Source (prototype)
IFC	Industry Foundation Classes
KBS	Knowledge Based System

MPP	Mass Production Principles
RNA	Ribonucleic Acid (translates DNA into proteins)
SEAM	Socio-Evolutionary AEC Market
SDP	Structural Design Product(s)
SE	Systems Engineering
SIS	Steal Ideas Shamelessly
UML	Unified Modelling Language

List of Figures

Contents

CHAPTER 1

Introduction

> "Instead of trying to produce a program to simulate the adult mind, why not rather try to produce one which simulates the child's?" - Alan Turing

One after the other shim design déjà vécu lampooned the author after a few years of practice as a structural designer in the Architecture, Building and Construction (AEC) industry. These sensations of reliving design problems, design processes and design solutions, raised the strong illusion of performing repetitive work that should have been identified and raided by automated Design Support Systems (DSS)[1]. Yet those DSS remained notorious automatons compared to human colleagues who actively adopted tasks.

The scantiness of adaptive capabilities in current DSS elicited attempts by the author to create models together with proper human computer interfaces, in which a wide range of recurring design problems could be solved. However, the author experienced that such isolated top-down efforts are

[1] DSS include software for Finite Element Analysis, Document Management, Planning, Risk management, Reporting and other software that contribute to solving design problems.

disproportional to their benefit. They required extensive and precise expert knowledge[2], their maintenance (e.g. adding parameters or implementing new code standards) regularly caused unwanted side-effects and end users remained sceptical about the outcome. Expert consultations in trusted man-to-man conversations often yielded more relevant and more reliable context-specific answers to design problems than extensive models with slick human-computer interfaces could.

It became obvious that design support cannot solely be improved with state-of-the-art models and human-machine interfacing techniques[3]. In the words of Beheshti (1993), "design decisions are largely influenced by experience, creativity, innovation and other factors" (tacit knowledge), which is difficult to catch in tight design systems (explicit knowledge). Therefore many aspects should be addressed when creating design tools: Which tools and which interfaces are the most appropriate in which situations? Should knowledge and intelligence be implemented within the tool or as a separate entity? Is it possible for a tool to communicate design intent with designers using natural expressions (e.g. subtle body language)? Could a tool become a designer itself and should it have a body? Is a virtual designer the best solution?

New déjà vécus lampooned the author during such attempts to improve current DSS, this time due to a lack of adaptive capabilities in programming environments which the author used to program new tools. Would it be necessary to improve on programming languages and environments themselves first before improving DSS? How many levels of meta-design[4] were to be addressed before adaptability could be incorporated in DSS and finally eliminate déjà vécus?

Human designers entertain that adaptability. For instance, designers may rely on sophisticated and specialised human “interfaces” such as qualified secretaries, modellers, structural engineers, specialists and document

[2] There are many definitions of Knowledge. Albert Einstein stated that "Knowledge is experience. Everything else is just information"

[3] The author created a design tool based on handwriting and drawing recognition (van de Ruitenbeek (2003)) and a 0.5m*1m multitouch table to explore how DSS could be improved.

[4] Meta-data is data about data, for example keywords in a document. Meta-design is design about design, for example frameworks that support the design itself (social networks, DSS, etc.). The author over-ambitiously created a meta-meta DSS by integrating a speech recognition system for a programming environment.

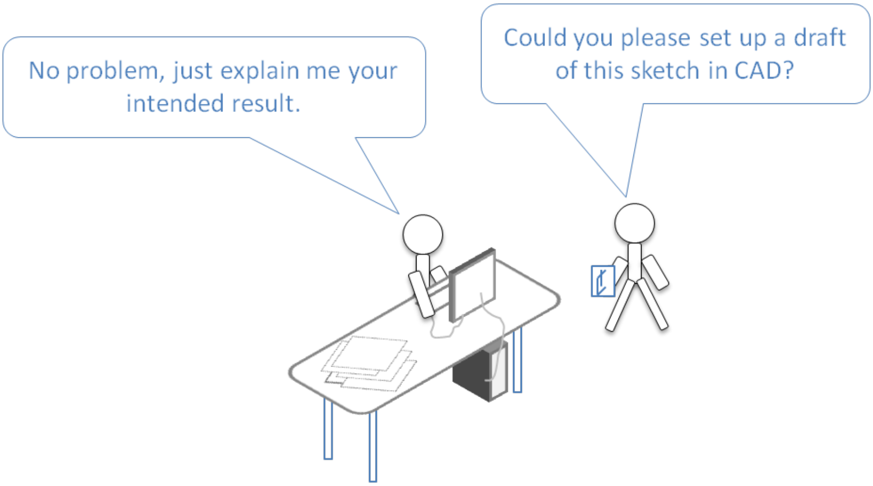


Figure 1.1: *An effective method to overcome human-computer interfacing issues is to employ human actors who are specialised in that task.*

controllers (Figure 1.1) to perform certain tasks. Werher von Braun stated something similar: "Man is the best computer we can put aboard a spacecraft... and the only one that can be mass produced with unskilled labour".

The following sections provide a systematic approach to the above incentive.

1.1 Problem description

The AEC industry employs numerous people (e.g the US counted around 5.6 million only in construction in 2010) in a wide range of occupations (such as construction workers, designers, managers, truck drivers, cost estimators, paperhangers, architects, engineers and lawyers). Special trade contractors (building finishing, foundations etc.) represented 64% of the construction industry, followed by 23% for construction of buildings and 13% for heavy civil engineering construction (BLS (2010)). As a whole, the AEC industry organisation is diverse, complex, distributed and decentralised (Dado (2002)).

Various institutions (e.g. Bouwkennis (2011)) annually investigate the waste[5] and its cause within the AEC industry. They report that the waste

[5] The waste here is defined as the unnecessary costs due to the process of translating cus-

is currently being estimated between 10% and 50% of the AEC industry's yearly overturn, and that it is being caused by:

- Insufficient consideration of designs' constructibility
- Inefficient communication & cooperation
- Lack of sharing knowledge and not learning from mistakes

AEC enterprises are attempting to eliminate these causes. The dominating means to do so in practice currently are (Davis (2007))

- Building Information Modelling (BIM)
  BIM is intended to ensure the integrity of all relevant project data for all involved actors in order to improve cooperation. An example is a 3D geometric model that is linked with planning, cost estimation and document management systems. Users get tailored information based on their specific role.
- Lean manufacturing
  Many of the aforementioned causes may partially be eliminated by bringing people of the various disciplines together in regular intervals. These "lean sessions" fraternise people, stimulate communication, focus attention on critical points, improve learning from mistakes and eliminate non-value-added production steps. Similarly, partnerships with subcontractors improve chain integration.
- Standardisation and improvements
  Frequently executed tasks can be automated and laid down in standards[6] to achieve a higher rate of knowledge reuse and more uniformity. Mistakes and successes are triggers to improve previously laid down standards and procedures.

BIM, Lean and Standardisation are inspired by mass production industries such as the car and ship industries. The governing assumption is that the application of mass production principles (MPP) will lead to waste reduction in the AEC industry as well. Indeed there are many mass production

tomer specifications into built products. If long-term efficiency, reuse, environmental impact and such are taken into account, then the numbers are much higher.

[6] Examples are predefined calculation sheets, working procedures and object libraries

situations in the AEC industry (e.g house blocks, standardised beams). Moreover, since repetitive work even occurs within one-of-a-kind products, MPP penetrated all layers and all divisions of enterprises. When it comes to true repetitive products or very generic tools, the application of properly managed MPP is a positive development.

Still, a large portion of the AEC industry activities is and arguably will remain one-of-a-kind. For example, the overall design of large bridges, special buildings and new roads are rarely being copied since the architectural design, environmental conditions and available budget are dissimilar for new designs. In addition, various competing forms of BIM, Lean and Standardisation started to emerge (the islands of automation phenomenon, Pollalis (1997)) since each enterprise, its subdivisions and its people have their own views on MPP.

While MPP yielded many valuable micro-level improvements, the ancient spaghetti model still dominates all levels - anticipating the next technological wave[7] (Figure 1.2). The fact that the AEC industry uses massive amounts of materials does not necessarily make it a mass production industry.

1.1.1 Accepting the seemingly chaotic organisation

One might expect a high level of uniformity and standardisation throughout the products and interfaces of 'genuine' mass production industries such as the smartphone, television and computer industries. Nonetheless, even those industries face a major amount of mutual competition, incompatibility, diversity and innovation (e.g. data exchange protocols, connectors and operating systems). Similarly, with the introduction of a growing amount of MPP, the AEC industry exhibits a natural resistance against uniformity and follows the rules of a free market in its full diversity[8].

Therefore the actual problem is the governing premise (almost the belief) that BIM, Lean, Standardisation or other useful means will lead to

[7] A recent example is the introduction of the new Eurocode. The Eurocode is intended to create a uniform approach to design, however each country uses its own national appendix which allows it to overrule the text.

[8] For example within the past five years, computer aided design, the use of systems engineering, document management, mobile devices and contract forms significantly changed the AEC work environment, making uniformity more difficult than ever. Also, various competing BIM and exchange formats appeared.

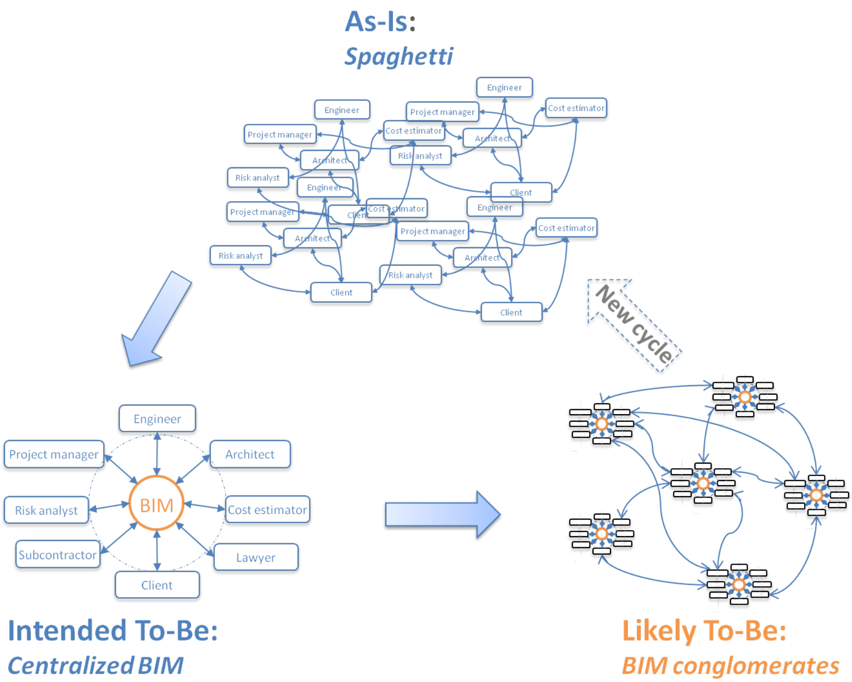


Figure 1.2: *Spaghetti to spaghetti cycle. Unifying efforts rarely lead to solving cooperation issues; rather they result in a certain degree of improvement since they are subject to competing implementations. For example, the current* as-is *spaghetti-like collaboration in the AEC industry was intended to be solved by a* to-be *centralised BIM, but it resulted in various* likely-to-be *BIM implementations (a process known as fission). This results in increasingly complex and diverging cycles (Figure 2.2).*

uniformity and ultimately to waste reduction. While this premise may be true for time-limited, genuine mass production situations, it should at least be put in perspective:

- Even though the AEC industry seems to be unregulated and suffers from waste due to incompatibilities, it survived many design code changes, economic crises, new technology and other disturbances, adapted and continued to produce buildings and infrastructure.
- The AEC industry is very unlikely to change into a uniform industry because of its diversity and the large amount of conflicting interests.

An additional premise therefore should be that the AEC industry's diversity is essential to achieve waste reduction on the long run[9]. The problem is that currently, there are no dominating means or developments that take advantage of the seemingly unregulated AEC industry organisation in order to reduce waste.

An appealing example of combining both premises comes from the car industry. A number of companies are experimenting with autonomous vehicles. In theory, if every car were equipped with a chip together with appropriate sensors, and if the road were furnished with marker chips, then computers could take over driving. Obviously, there would be much discussion about the technology to be applied, security and privacy issues, uniformity and such, and high cost would be involved in maintaining the system. Alternatively, the Google autonomous vehicle (Pinto (2012)) partially uses markers (such as GPS and maps) but also heavily relies on real-time, real-world visual information. The latter positively influences the price and reliability of the system, and eliminates the need for every pedestrian or other moving objects to carry some marker.

1.1.2 Scientific knowledge gap in the AEC industry

Currently, an overwhelming amount of research in waste reduction in the AEC industry is dedicated to BIM, Lean and Standardisation. The reason might be that these MPP are relatively easy to implement, simple to explain and can be measured objectively. For example, nobody would object a useful calculation sheet that outputs the exact amount of reinforcement in a beam given the dimensions, conditions and forces.

Much less attention is devoted to handling the natural fission process after a new model (e.g. the sheet) has been introduced. What should an enterprise do with the original if another employee introduces a better sheet for specific conditions or if design codes change? Who is responsible for the sheet's validation and how can its quality be measured? What can be done to prevent valuable ad-hoc created calculation tools from disappearing? How can several tools be tied together? How can the appropriate tools be selected for a specific design task? How can knowledge[10] be man-

[9] If you cannot beat them, join them.

[10] Rezgui et al. (2010) lists three levels of knowledge management (KM) within the AEC industry. The first two generations (simple document-based knowledge sharing & knowledge conceptualisation and nurturing) involve a continuous struggle to store,

aged and reused in daily practise within a seemingly unregulated industry and without disproportional effort? Kamara et al. (2002) argue that "effective knowledge management requires a combination of both mechanistic and organic approaches in an integrated approach that incorporates both technological and organisational/cultural issues".

Even though it may be done more efficiently, human designers have such capabilities; both as individuals and as groups. Lang (2002) stresses that "effective design support systems must complement human cognitive activities" in order to prevent the loss of design intent. Sketches, human language and such contain highly relevant design contributions that are rarely included in current design support systems as it is extremely difficult for computers to represent and interpret them in a meaningful way (Horváth and Rudas (2009)). Designers on the other hand can interpret real-world situations, select and use the appropriate knowledge, apply the various MPP, use each other's expertise and synthesise a valid design. Cross (1982) believes "design develops innate abilities in solving real-world, ill-defined problems". The ability to mimic such qualities in a designerly tool is a knowledge gap in the AEC industry that the author will attempt to explore.

1.1.3 Research focus

Waste is being produced during various stages of the building process: design, construction, maintenance, demolition, reuse etc. The previously mentioned annual reports indicate that a large portion of the waste (approximately 60%, depending on the type of enterprise) can be prevented during the early design stage.

Due to the decentralised, multi-disciplinary and asynchronous nature of the AEC industry design process, it is difficult to pinpoint when this important early design stage happens. Arguably, this happens at the start of any design activity, whether small or large, during the project inception or at the detailed engineering stage, and regardless of the practitioner's role. Any design activity starts with initial crucial choices; it is the designer's speciality to know what to choose within a specific context.

Therefore the focus of this research is on the early design stage, in a

---

retrieve and maintain knowledge in the form of documents or concepts. The third generation integrates in real-life rather than prescribing KM (much like the autonomous vehicle project in Section 1.1.1)

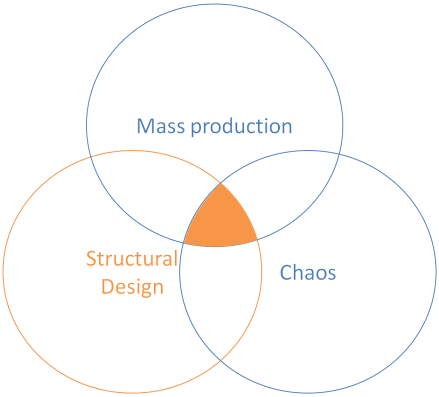


Figure 1.3: *The research focus is on the relation between mass production principles, the seemingly chaos within the AEC industry and structural design within the Netherlands.*

seemingly chaotic environment in which mass production principles play an important role. This will be narrowed down to the context of structural design in the Netherlands (Figure 1.3) due to the author's own field of expertise, experience and education.

1.2 Research questions

The key elements in the problem description, scientific knowledge gap and focus of research may be bundled in the following research questions:

1. How could the advantages of mass production principles be combined with decentralisation and non-uniformity in order to effectuate waste reduction in structural design?

2. How could knowledge management be integrated in daily structural design activities effectively and without disproportional effort?

1.3 Hypotheses

Analogies with other systems in which knowledge, MPP, chaos, competition, being social and such play an important role, may provide useful insights that could lead to answering the research questions. Examples are the free market, the Internet, the human brain and social communities. The governing scientific theories about such systems are available in for example cognitive science, complex systems, fundamental physics and the evolution theory[11]. Based on these analogies, the author proposes the following hypotheses:

1. Principles of cognitive science, complex systems and the evolution theory provide insights into effective methods of achieving waste reduction within the context of structural design activities.

2. Extremely easy to use Human Machine Interface (HMI) will yield to more efficient use of AEC tools and products.

The elements of the hypothesis may be put together in the form of an imaginative superior being, named Levente[12]. It is omniscient in structural engineering, flawlessly interacts with humans, computers and its own species, never makes mistakes twice, never becomes tired, resolves conflicts, generates creative, viable solutions and is physically unbound. With its unlimited computational power, it invents perfect protocols, formats and software to which all actors in the AEC industry fully comply. Finally, within a changing environment it brings forth adapted offspring.

In a first attempt to create a simplified, realistic version of Levente, the author researched literature on robotics and artificial intelligence. Since the ability to interpret the real world visually (e.g. information on a computer screen, a building site or a written note) is of primary importance for a designer, the author focussed on Levente's visual system. This resulted in a number of prototypes that were intended to allow Levente for interpreting the real world visually (Sections A.3, A.4).

From these prototypes together with the advise of robotic experts, the author acquired a better understanding of the immense complexity and interdependencies that are involved in producing Levente. Other researchers accomplished creating humanoids[13] with remarkable versatile capabilities such as advanced visual systems, body control, autonomy and language understanding (examples are NASA's Robonaut, Burford and Blake (2001),

[11] Even though the evolution theory is not entirely proven, it is a recognised field of science that helps to explain complex systems.

[12] Hungarian male name, meaning "existing", "being"

[13] Human lookalike in form and / or function.

Kim et al. (2004)), but none of these are yet capable of operating in the same manner as a human designer in the real world[14].

Even so, it may be a matter of time before technological advances are far enough for Levente to materialise (similar to the sudden, widespread introduction of mobile devices[15]) - forcing enterprises to include it in their new strategy[16]. Apart from the technical challenges, it should be kept in mind that a (superior) copy of a human being would inevitably introduce new mistakes & compatibility issues but also will cause difficult debates about responsibilities, reliability and ethics. Meanwhile, the author will attempt to propose a realistic theory that fits the current trend of evolving technologies.

1.4 Research methodology

The objective in this thesis is to provide a theory for improving design support that structural designers or researchers could use in order to achieve waste reduction. A literature survey will be carried out to formulate a preliminary version of this theory (Section 2) by using analogies with other fields of science (as described in the hypothesis above). The required type of analogy reasoning is metaphorical (Vosniadou and Ortony (1989)), since the available examples within the AEC industry are scarce and incomplete. The analogy reasoning will not be pushed extensively nor will formal transition methods be applied. Rather, the metaphors will be used as a source of inspiration.

For the purpose of proving certain aspects of the theory, a prototype (3) will be designed and distributed amongst structural designers. The collected data will be used to gradually improve both the theory and the prototype in a number of cycles. Rather than describing this entire process, the content of this thesis will be limited to the final version of the theory and prototype.

In order to ensure that end users' feedback includes the prototype's technical performance as well as their product experiences, the author will

[14] Figure 2.3 shows the difference between the noumenal and phenomenal worlds.

[15] According to Sybase (2011), 70% of the world population in 2011 already owns a mobile phone. Some countries like Italy have a penetration rate of as high as 148% Gruber and Koutroumpis (2010).

[16] Anticipating on robots in everyday life, Clarke (1994) formulated laws such as "Robots may not injure humans or, through inaction, allow a human to come to harm"

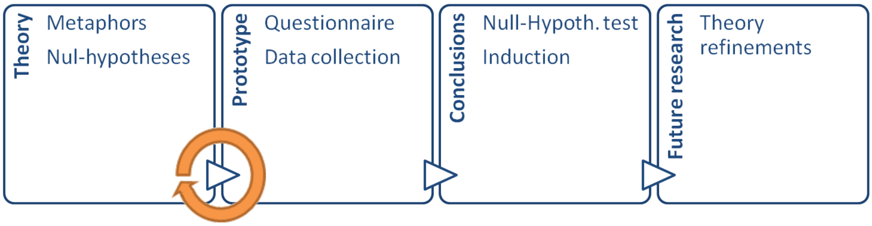


Figure 1.4: *Research Methodology. Metaphors will be used to build a theory, followed by null hypotheses. The theory forming and prototyping process is iterative. In the conclusions, the null hypotheses will be tested against the evidence gathered from the prototype. Induction of the theory will follow and suggestions will be given for future research.*

use both qualitative and quantitative methodologies (e.g. collecting data and conducting a questionnaire). Since the world of structural design is very divers, the prototype will be created on a platform that is widely accessible (such as the Internet).

With the limited prototype, it may be difficult to objectively measure whether the new theory contributes to waste reduction compared to the current situation. Therefore testable null hypotheses will be formulated after constructing the theory.

The external validation of the theory will constitute of testing the prototype involving a large number of structural designers.

Finally, the author will present the conclusions and define areas for future research. Since the prototype is unlikely to cover the entire theory, researchers and enterprises will be encouraged to further test and optimise the theory with their own prototypes.

CHAPTER 2

Evolutionary, unconscious design support

> "It is not the strongest of the species that survives, nor the most intelligent. It is the one that is the most adaptable to change" - Charles Darwin

THE first step in this chapter is to metaphorically compare the AEC industry to a complex adaptive system (Section 2.1). This analogy provides a scientific meaning to the seemingly chaotic AEC organisation (Section 1.1.1) and a guidance to derived analogies such as the collective mind and evolution. The objective of this chapter is to provide a theory with which designers and future researchers may use to improve design support. Null hypotheses will be formulated in order to measure whether applying the theory in practice has any effect.

2.1 Complex adaptive systems (CAS)

The AEC industry largely depends on human actors. In spite of an increasing level of automation, there would be little or no activity if all human actors laid down their work. Humans created and formed the AEC industry over the centuries into a system that is capable of solving "wicked"[1] problems. They divided it into various disciplines (e.g. design, construction, offshore etc.), generated spin-offs (e.g. new, independent specialised enterprises), scattered over the entire world[2], built social networks, created theories and imposed design codes to which they committed themselves.

This productive, changing and self-regulating industry matches well with the description that "... complex adaptive systems exhibit true self-cause: parts interact to produce novel, emergent wholes; in turn, these distributed wholes as wholes regulate and constrain the parts that make them up" (Juarrero (2000)). Therefore the AEC industry has the characteristics of a CAS. They are "turbulent environments where change is imminent and frequent" (Dooley (1997)), "the intermediate between perfect order and perfect disorder" (Tiezzi (2006)) in which "conflict is one of the most important social factors shaping the evolution of living systems" (DeDeo et al. (2010)). Individuals within these systems create and maintain the system and in turn, the system protects and regulates the individuals for the collective benefit (Section 2.2.4). CAS are often living systems[3] such as biological and organisational systems (Paperin et al. (2011))

Literature on the analogy between the AEC industry and CAS is scarce; many authors consider the AEC industry to be a very complex and chaotic system, and consequently reside to BIM in order to reduce complexity and to bring order. An exception is Benne (2006), who compared AEC project organisations with living CAS and argues that complexity in AEC project organisations should be accepted as a fact that is unlikely to change, and that adaptivity is the key factor to waste reduction.

The AEC industry as a whole and structural design in particular fulfil the following three criteria for being a chaotic system (Hasselblatt and Katok (2003)), even though they may seem stable for long periods of time:

---

[1] Wicked problems (a term coined by Churchman) are problems that have complex interdependencies and incomplete, changing and contradicting requirements. “Each attempt at creating a solution changes the understanding of the problem” (Rittel and Webber (1973)).

[2] There are plans to build structures on other heavenly bodies, e.g. moon bases with D-Shape, a 3D printer that uses sand and a binder.

[3] There is a close relation with Artificial Life. “Artificial life embraces those human-made systems that possess some of the key properties of natural life." (Taylor and Jefferson (1993))

1. It is sensitive to initial conditions:
   Consider a project or design process start-up. Small changes, such as the project members' moods, the initial requirements or the available information largely influence the process as well as the final outcome.

2. There is topological mixing[4]:
   The sets in the phase-space of a project (i.e. the possible states and their possible state transitions) eventually overlap. For example, a project may be successful but evolve into a disaster, while the project may also start as a disaster, be saved by an innovation and end up successfully. These transition-sets eventually overlap.

3. The periodic orbits are dense:
   The system (e.g. a project) generally follows a certain procedure (orbit). New projects are intended to follow that same procedure (periodic), however there are always small disturbances along the way, which causes slight differences in the orbits (they are dense) - even if the initial conditions are identical. Some disturbances may cause the orbit to follow an entirely unexpected path (causing topological mixing).

The AEC industry is not only a complex[5] chaotic system but also a CAS since it is unpredictable but at the same time anticipatory due to its learning agents (i.e. the structural designers). Two important CAS aspects are:

1. Its behaviour is difficult to be explained from its constituent parts: it is emergent (Holland (1999), Goldstein (1999)). The human mind, culture and the way in which structural designers develop their knowledge domains are exemplary.

2. It adapts to its environment in an evolutionary way (Levin (1998)). Loorbach and van Raak (2005) nuanced that especially in policy research, not only Darwinian evolution[6] happens, but also punctuated

[4] Much like Murphy's law: "If anything can go wrong, it will".

[5] Weaver (1948) distinguished that a complex system may be disorganised (e.g. mixing fluids) or organised (with emergent regulatory parts, such as happens in the AEC industry - e.g. spin-offs and design codes).

[6] Aside from genetic manipulations, evolution is undirected and irreversible.

equilibrium[7] (such as a sudden introduction of the Eurocode). Another special case is genetic manipulation.

Over-stressing one of the CAS aspects (e.g. standardisation or chaos)[8] and imposing it on the AEC industry is therefore likely to fail. It must be accepted that the AEC industry as well as structural design are CAS that require a proper balancing of all CAS aspects.

There are two main challenges for translating the two above aspects into the context of structural design support:

1. Allowing for a collective mind (metaphor: Theatre of Consciousness, Section 2.2):
   (a) Support for regulated cooperation: the conscious.
   (b) Support for distributed, diffuse and implicit cooperation (such as communicating with various dissimilar actors) without having commonly agreed data exchange formats: the unconscious.
2. Allowing for product evolution (metaphor: evolution in nature (2.3):
   (a) Support for repetitive, rather static products (such as recurring construction elements) by using evolutionary development and structuralism - similar to evolution of organisms with genes (Section 2.3).
   (b) Support for average products (such as a preferred type of viaduct), again by using evolutionary development and structuralism.
   (c) Support for one-of-a-kind products (such as unique buildings), by allowing for species to 'suddenly' appear in the evolutionary environment (e.g. through genetic manipulations (innovations[9])).

[7] This theory (Eldredge and Gould (1972)) states that there are sudden changes in species (e.g. due to a catastrophe that changes the environment) followed by long periods of stability.

[8] Standardisation is useful but it is just not sufficient for improving design support as a holistic concept. An example is over-stressing the importance of BIM. Perhaps BIM should be complemented with Agent Based Modelling and Simulation.

[9] Leifer (2000) describe innovations much in the same way; an incremental innovation focuses on feature or cost improvements while a radical innovation is “a product, process, or service with either unprecedented performance features or familiar features that offer potential for significant improvements in performance or cost”.

From that moment, they are subject to normal evolutionary developments.

Note that applying evolutionary algorithms (e.g. genetic algorithms) to solve and optimise complex design problems is not new (e.g. Ritzel et al. (1994)). However in this thesis, product evolution will be viewed differently. It will be used to follow a product's evolutionary developments after it has been applied in a project. This happens after designers copy and mutate it. Designers (instead of computer algorithms) will be regarded as a live 'genetic algorithm'.

For the purpose of mathematical computations, a CAS is often modelled as communicating nodes - e.g. a directed graph or cellular automata. The nodes within these models only follow simple rules - i.e. it is not complicated. For a CAS with human 'nodes' (agents, such structural designers in the AEC industry), it may be different since humans can reason about their own CAS and to a certain extent oversee or manipulate the situation. Therefore humans do not only expose mechanistic adaptivity, but also apply 'genetic manipulations' on the CAS.

From this observation, the author investigated literature on the human brain, quantum mechanics, biology, psychology, cognitive science and philosophy and noticed that CAS embed abstract aggregations in segmentations, of which DNA is the best example (Appendix C). This inherently distributed, redundant and extremely implicit form of incrementally emerging knowledge in a social environment may be a suitable analogy for AEC industry knowledge management.

The following sections elaborate on the collective mind and product evolution.

2.2 The collective mind

This section intends to harmonise total organisational order (e.g. predefined procedures) with total chaos (e.g. creativity). Both extremities exist in a typical CAS. The question should not be whether to strive for the one or the other, but rather how they should be balanced optimally within a specific context. The resulting collective mind largely dictates the effectiveness with which designers approach a problem. In this regard, an analogy can be the human brain or any other organisational structure in which there is order and anarchy, collectivity and individuality - such as an ant colony.

2.2.1 Theatre of Consciousness

The human brain is a combination of the conscious and the unconscious. The conscious addresses things that can be overseen and that can be reasoned about. The unconscious is capable of solving problems with a high degree of complexity, from the point where the conscious has difficulty to oversee them (Dijksterhuis (2008)). The unconscious is estimated to be 200.000 times more powerful than the conscious (ibid). Its organisation is seemingly chaotic but extremely efficient.

Neurophysiologist Libet et al. (1993) proved that the unconscious decides on actions before a subject consciously becomes aware of it. With his findings, the concept of an active, free will had to be depleted and the conscious degraded to a simple spectator with the illusion of being in charge. Based on that outcome, Baars (1997) compared the brain with a theatre[10] as shown in Figure 2.1. The conscious is the podium while the unconscious is caught in complete darkness and continuously reacts on stimuli from neighbouring cells[11]. It operates in more or less independent modules. Despite of this complex system, coordinated actions are possible since the unconscious uses the podium to receive and publish information.

The Theatre of Consciousness matches the AEC industry organisation. Both are distributed and seemingly chaotic by design and yet they allow for cooperation and complex, meaningful behaviour by using an interdependent split between the conscious and the unconscious:

- Conscious:
  Podium (e.g. management layer) that is meant for steering and feedback. This management layer may seem to be in charge but in fact their actions are a result of preparations by the team members "in the dark".

- Unconscious:

[10] These ideas overlap with Minsky (1986).

[11] Several researchers (e.g. Dotta et al. (2011)) have measured photon emissions in brain cells, and suggest that light and quantum effects may form wireless cell communications.

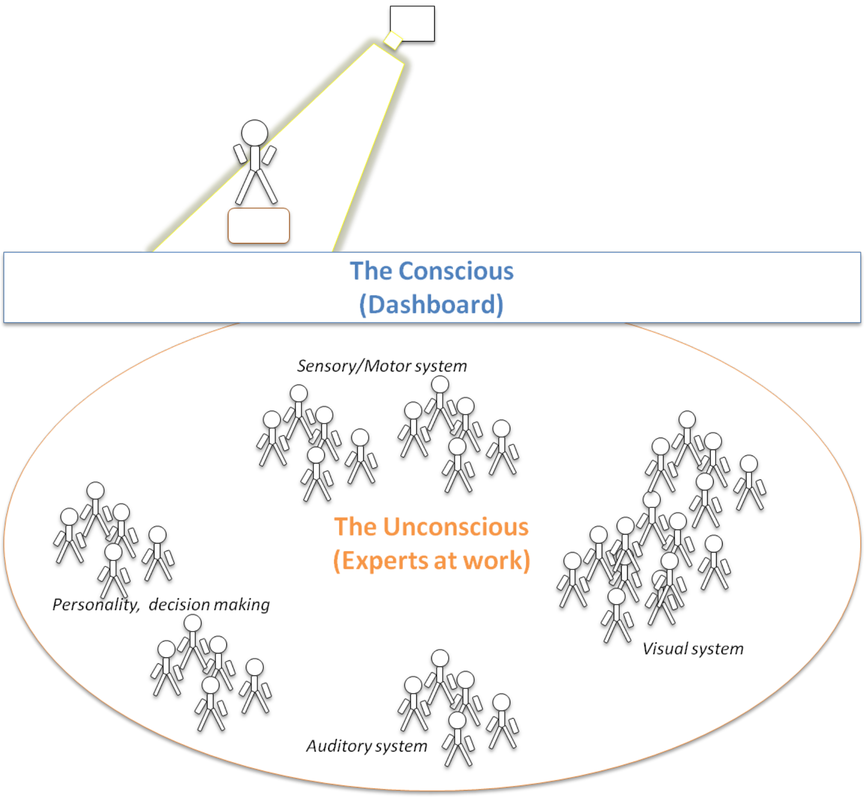


Figure 2.1: *Based on experiments by Libet, Baars proposed that the brain is a theatre. The conscious is the light-shedded podium, which is visible to the unconscious. The unconscious is caught in complete darkness and operates in more or less independent modules. Still, cooperation is possible since the unconscious receives and posts messages on the podium. [adapted and interpreted from Baars]*

> Many actions in the AEC industry seem to happen without any explicit, centralised steering[12] "in the dark". Yet, by having numerous connections with other team members together with the emergency signals from a management (and therefore in fact emergency signals from colleagues), people in the unconscious are the driving force of an enterprise.

Unsurprisingly, many authors reside to forms of the conscious in attempt

[12] Jung (1936) coined the term "collective unconscious". The complexity of the distributed, multi-actor AEC industry is in line with the saying in Bennett (2007): "If you work on one neuron, that's neuroscience. If you work on two neurons, that's psychology."

to reduce waste since it seems very logical to apply en mass. If only all designers would conform to a certain set of rules that eliminates chaos, then cooperation would improve and waste would be reduced[13]. A few examples of such initiatives in Europe are the Eurocode (Eurocode (2012)), VISI (CROW (2012)), COINS (CUR (2008)) and CHEOBS (CROW (2008)). However, all of them are subject to variations and competition. While they may be successful on the short term, inevitably there will be customisations of such rules to better fit a specific context. Soon enough such varieties become unmanageable and result in a new form of chaos and/or in a very rigid management body (as is the case with the Industry Foundation Classes (IFC) organisation). The reason for this standardisation spiral (Figure 2.2) to appear is that the conscious cannot possibly conceive all of this information - therefore it will either attempt to eliminate complexity or it will rely on the unconscious.

One way to overcome this issue is to enlarge the conscious - more managers, more regulations, more control. Another, more natural way, is to acknowledge the unconscious and to use it more efficiently. Currently, structural designers do dispose of an elementary form of the unconscious (e.g. their own networks and search engines), but there are very few tools that support the full Theatre of Consciousness in their daily work[14]. While there is much effort in the direction of the conscious (e.g. BIM), the unconscious is still underestimated and undervalued and therefore the author will focus on it.

Improving the unconscious may for example be achieved by following a bottom-up strategy:

- Case Based Reasoning (CBR, Section 2.2.2) may be used to allow designers for easily finding suitable historical solutions to specific design problems[15].
- Virtual assistants (Section 2.2.3 and A.4) are humanoids that should

[13] One may compare it with an attempt to revert democracy to a monarchy, and to subduct freedom of speech in exchange for censure.

[14] The symbiosis between humans and artificial intelligence systems is known as Ambient Intelligence (AmI). Several organisations are applying it to the AEC industry: OpenStructures ("everyone designs for everyone"), OpenDesign (Collaborative design environment) and MIT's Open Source Building Alliance Operation (MIT (2011)).

[15] This has a close relation with product evolution (Section 2.3), since product evolution is needed to facilitate the cases in the first place.

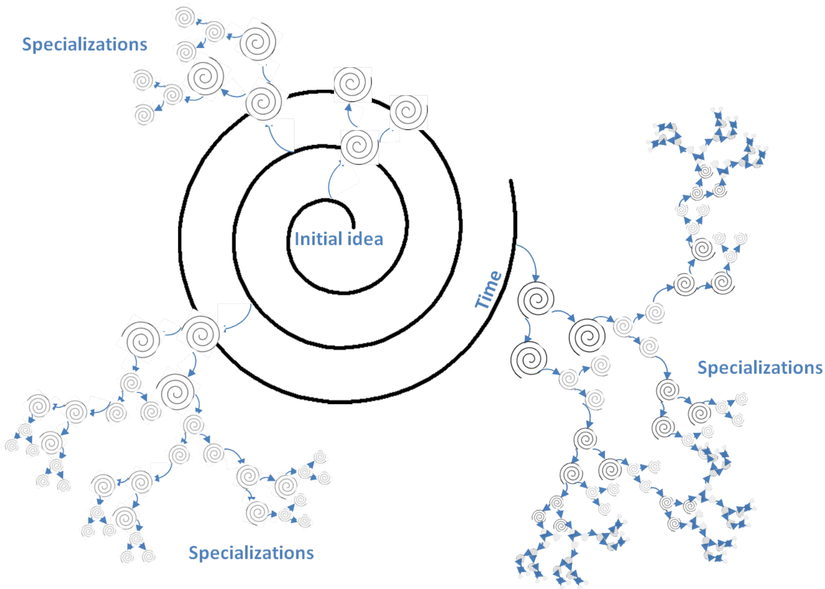


Figure 2.2: *The design process starts around initial ideas that grow and improve over time, especially if there is feedback from actual built objects. Due to Re-presentations, derived ideas and specialisations inevitably appear and follow similar spirals. Some spirals might prove successful while others might disappear. The seemingly chaotic and distributed process of trial and error produces numerous variants. It has similarities with reproduction and natural selection in nature.*

be capable of real-world interaction (e.g. being able to understand a sketch, a designer's instructions or working with computer programs' normal graphical user interfaces) specifically in context of structural designers (e.g. using CBR). This eliminates the need to agree on explicit, formal formats/protocols for computer processing.

- The Leviathan (2.2.4) is the environment in which designers and virtual assistants operate. It provides the means and the rules, and is very similar to the Theatre of Consciousness.

These three topics are explained in more detail within the following sections. Arguably, it is almost impossible to set up such complicated networks in the classical way (i.e. with the purpose to solve design problems) for structural design, let alone the AEC industry. Therefore an alternative can be to regard the existing set of structural designers together with their connections as an existing, working neural network (i.e. the much-disputed spaghetti model) that should be improved, stimulated and expanded.

2.2.2 Case-Based Reasoning

Case-Based Reasoning (CBR) is a well-known technique for solving new cases based on historical cases, similar to the way in which humans apply heuristics. The CBR founder originally coined it a "dynamic, evolving memory structure" (Shank (1982)). As described by Weber et al. (1996), CBR is the artificial intelligence implementation of similarity heuristic. Rissland (2006) describes that AI has achieved much, but it still cannot “deal well enough with the inherent messiness that characterises much of the world that humans an AI artifacts operate in” and suggests to further investigate similarity-driven reasoning, analogy learning and explanation.

Two prerequisites for a properly functioning CBR in the context of structural design are formulated below. As soon as both prerequisites are fulfilled and structural designers start using CBR systems en masse, the number of available Structural Design Products (SDP, such as simple spreadsheets or complex fly-over designs) will automatically increase and ensure statistical significance.

1. All relevant SDP must be published in the CBR system in an extremely easy, in-place[16] manner. Enterprises must decide on whether or not to make SDP public and whether or not to query a selection of remote CBR systems. A multiple of competing vendors are likely to launch their own CBR brands (e.g. specifically for structural design of long bridges in Europe).

2. It should be extremely easy for structural designers to copy SDP. That will arguably stimulate the use of existing SDP at a larger scale, and SDP that were assumed to be one-of-a-kind may prove to be repetitive. Since a SDP is always embedded in a certain context, one way of finding relevant SDP (similarity search) is by using context matching.

A number of problems must be addressed for such a CBR system to work:

- What stimulus does a structural designer have to upload his designs? The smartphone app markets have shown that developers are willing

[16] In-place systems are non-intrusive; they assist users in a fully opaque manner.

to produce and publish useful apps in return for revenue, fame, a large and responsive user base to be proud of etc. Similarly, AEC enterprises could stimulate their employees to publish their work in an "AEC SDP market".

- What stimulus does a structural designer have to download designs?
  If a structural designer downloads a fellow designer's work, he gets access to a well prepared design. He knows whom to turn to for additional questions, new features, bug fixes and such, and he may add that person to his own network. CBR might reduce the need for performing lengthy calculations; with a sufficient amount of relevant SDP, filling in a few parameters is enough for generating preliminary results.

- How could the quality of uploaded designs be measured?
  There may be irrelevant SDP, "test" SDP, derived SDP (i.e. designed exclusively with CBR without additional calculations), cases with hidden errors (diseases). The smartphone app market mechanisms provide mechanisms for eliminating such SDP (immune system). For example, ratings, comments and the number of downloads may indicate a specific design's popularity. The system should additionally indicate in which projects and by whom the SDP were copied, so the user could find even more context-specific information. As John Hopfield (neural network builder) said, "Biology, by and large, is not interested in finding the best things, just things that are pretty good that can be found quickly."

- How much work is it to upload/download designs?
  This is one of the most important questions. If it takes too much effort to publish or to copy designs, then the system will fail. There may be two approaches to address this issue.

  - Similar to the smartphone industry, one may create a specific operating system and associate an ecosystem with it. Structural designers will be required to create their designs in a specific, compatible way.
  - One may attempt to relief the structural designer of the complexity involved with uploading/downloading designs. Instead, an

intermediate system could extract and propose relevant designs. Again, there are two options (and a combination of both):

* The intermediate system is a plugin (Appendix A.1) that integrates into structural designers' tools (e.g. their FEM and spreadsheet software). The plugin extracts and proposes relevant information in-place while hiding technical CBR details. For example, if a user types "cofferdam" in a search box then the plugin may show historical and certified SDP. When he fills in the values (e.g. type of soil) then statistics could indicate how well his input and output match that of other designs. Finally, if he finished the calculations then his work becomes a new SDP.
* The intermediate system is a virtual assistant (Section 2.2.3). The assistant uses human interaction techniques (speech, vision etc.) to determine what the structural designer intends to do. In the background, it queries the CBR system(s) as to gather suitable advise for the human designer. The advantage of a virtual structural designer is that it is not restricted to specific software applications (like a plugin) but operates within the human perceptual world.

- What about standardisation[17]?
  The need for explicitly maintaining SDP libraries may be eliminated. As soon as the CBR system finds that a certain SDP is popular, then it may trigger a knowledge manager to validate the product and to certify it[18]. The certified SDP appears as a preferred SDP on the CBR market, but it will remain subject to the evolutionary rules. Should there be new design codes or context changes, then that SDP could drop in popularity, perhaps in favour of a new SDP. In this way libraries with standards may be managed bottom-up and stay up to date.

Currently, CBR is unable to replace human structural designers. Still, little

[17] "If you think of standardization as the best that you know today, but which is to be improved tomorrow; you get somewhere" - Henry Ford

[18] Several techniques such as attractors in chaos theory and building blocks in genetic algorithms may be helpful to identify such popular SDP.

is known about the personal mental process that structural designers follow to make sense out of chaos. Kolko (2010) observed that designers "immerse themselves in a particular subject or discipline, then go 'incubate' that material. After a period of reflection they will produce a tangible artifact". He also mentions that the only measurable thing about the design process is the input and the output, not the synthesis (although there is effort in capturing the synthesis, also known as design rationale, as well (Fischer et al. (1991))). It may suffice as a start for the CBR system to limit the capturing to the input and output. It will help other structural designers to find relevant SDP.

When applying CBR for structural designers, the starting point should be to a) make it extremely easy to use and b) to eliminate the need to confine to standards. This is the subject of the next section.

2.2.3 Virtual assistants

DSS (including CBR) have their specific functionality and limitations, and therefore designers use a personalised variety of DSS. Consequently they must enter data multiple times and in different formats. For example, the outcome of a FEM calculation is often used as the input for concrete section spreadsheets. Many attempts have been made to improve this situation. For example, DSS were extended to include a plethora of integrated tools, Service Oriented Architectures abstracted DSS input and output in high-level interfaces and BIM developers attempted to interconnect project information through predefined standards. Still, structural designers are not entirely served by these solutions. The new high-level tools may still not be of use to them, the integrated functionality may be difficult to find, or they may simply have more faith in their own tools.

Levente is intended to relieve structural engineers from the complexity of back-end systems (such as a DSS and difficult to use tools) whenever possible (Appendix A.4). Levente is not omniscient nor is it a replacement for existing software or humans. Rather, it hides complexity from users (e.g. to copy paper-based information into a computer application) by being in the same perceptual world as designers (Figure 2.3) and by entertaining multimodal humanoid conversations (Knudsen and Brainard (1995)).

Creating Levente implies copying a human (at least as an abductive

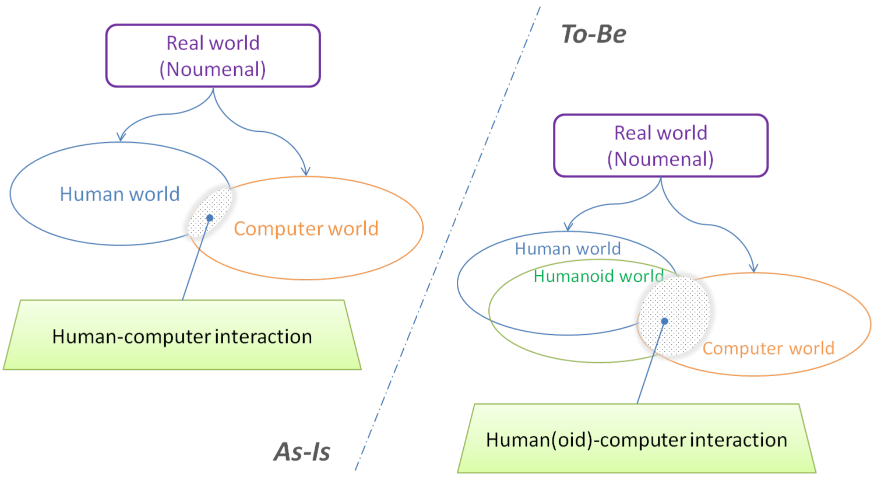


Figure 2.3: *As-is and To-be noumenal and phenomenal AEC worlds. The real world appears differently to humans and computers and the overlap in both worlds represents human-machine interaction (HMI). The humanoid Levente is intended to maximise this overlap in order to gain an optimal understanding of humans and the problems they face in the AEC industry. His overlap with the computer perceptual world does not need to exceed the normal human-computer interaction area since he can use existing HMI.*

mindset) but it is extremely difficult to do so (Section 1.3). Nevertheless, there are simpler existing 'creatures' such as smartphones that are capable of answering spoken questions and of providing information about objects that are exposed to their cameras. Piece by piece, Levente may materialise in the form of increasingly advanced smartphones, contact lenses (augmented reality), implants or as separate humanoids that share the perceptual world of humans (Figure 2.3). Virtual assistants like Levente will likely be created in other fields of science and be implemented in the AEC industry, where they will rely on domain-specific knowledge (e.g. the structural design CBR system or even a custom education) to be of use.

It is conceivable that eventually, conglomerates of virtual assistants will emerge (Dautenhahn (1995)). Being able to communicate in natural language and sharing perceptual worlds makes them compatible with other virtual assistants and humans. Shuji Hashimoto, director of the humanoid robotics centre at Waseda University in Tokyo states that current robots need "kansei" which is Japanese for a raft of emotional notions such as feeling, mood, intuitiveness and sensibility; "... we don't have to be shy in im-

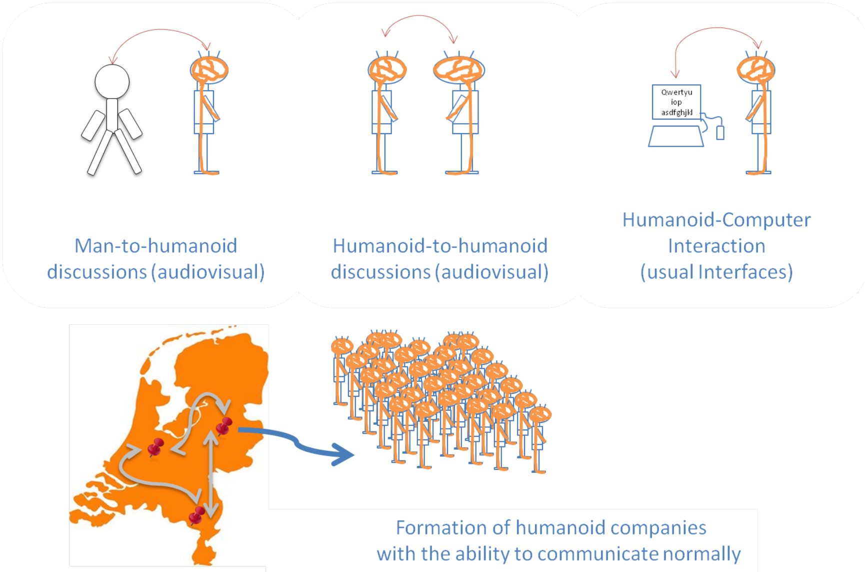


Figure 2.4: *Levente is not an omniscient supernatural agent; rather it is intended to be a learning, skilled humanoid colleague that is capable of interpreting the real world from a certain point of view and finding answers to problems through loose coupling in existing, tightly coupled external systems using visual and audio information only. Levente is not bounded to a physical form yet it needs access to visual and audio streams and be able to respond visually or verbally.*

plementing irrational mimetic behaviours in our robots" (Salvador (1998)).

The blended world of Leventes and humans will have an impact on the AEC industry in general and on structural design in particular. The AEC CAS will become an even more complex and chaotic Leviathan.

2.2.4 The Leviathan

In "The Social Contract", Rousseau (1947) opens with "Man is born free, yet everywhere we see him in chains". The Leviathan was a superior creature with whom one could exchange one's own freedom for protection (Hobbes et al. (1969)). Leviathan was Hobbes' metaphor for a social contract (e.g. an insurance); something big that exists for the benefit of the masses.

In a similar way, AEC actors collectively and evolutionarily crafted the AEC CAS. Gradually, the AEC CAS became a Leviathan in which designers became embedded rather than being in charge[19] and independent - for example, designers are now required to closely follow design codes that did not exist previously. Like RNA molecules, designers are not individuals any more (although individual viruses[20] will always exists) but messengers who translate design knowledge into instantiated designs within the context of a complex project.

It is difficult to think of the AEC industry as one all-encompassing Leviathan. Rather, there is a multitude of overlapping Leviathans (e.g. enterprises, branches and departments). Each SDP may be regarded as a small Leviathan as well, whose DNA is known (SDP knowledge), whose mRNA roles are fixed (i.e. dedicated expert structural designers), whose manufacturing and transportation processes are prescribed etc. Still, these small Leviathans are not true mass production units but CAS that are subject to an evolutionary environment. This will be elaborated upon in the next section.

2.3 Seed centred design

In describing the complexity of optimal functional designs, French (1994) writes that "... complexity in itself does not appear to be expensive in nature, and her prototype testing is conducted on such a lavish scale that every refinement can be tried. An engineer always has to balance manufacturing cost against performance ...". The difference between nature and structural design is that the latter (currently) has little playground to experiment. The prototype in section (3) is devoted to enlarging this playground. Note that even if both designers and nature have a large playground, the 'experiments' are still a matter of life and death (e.g. for a designer, a failed experiment might come at the expense of his job or a large financial consequence for the enterprise).

The following paragraphs describe this analogy between nature and the small Leviathan. First, seed embodiment will be discussed, followed by a view on the DNA within a seed. The last paragraph describes that DNA encompasses the whole instead of being a further decomposition.

[19] Like the poet John Dryden said: "We first make our habits, then our habits make us."

[20] A virus is not always negative. In the past, viruses caused mutations and spread sequences which caused genetic diversity. Today, viruses are being used for gene therapy.

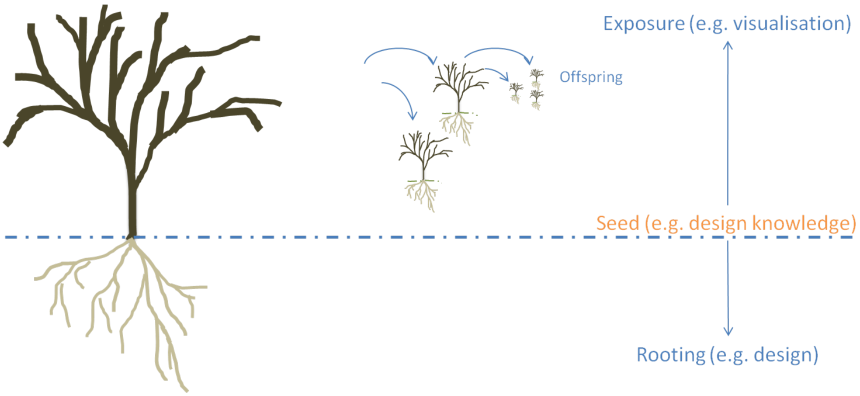


Figure 2.5: *A structural design product is often conceptualised around initial ideas (seeds). If nourished (supported) well within proper conditions (context), it will grow, root and expose; they might even produce offspring.*

2.3.1 Seed embodiment

Consider a seed as being an efficient and compressed module with DNA and an amount of initial nutrition from which ultimately a mature SDP roots and exposes itself to an extent that depends on the nourishment and environmental conditions (Figure 2.5). Having a start in the form of a previous generation's DNA[21] (the seed) and an initial amount of nutrition (e.g. the original structural designers' advice) is not enough. Arriving at a mature SDP type requires a process of trial and error and the ability to continuously and redundantly adapt and reproduce in an evolutionary environment (Nonaka and Takeuchi (1995)) - much like a computer driven genetic algorithm, except for that the nodes and the network in this thesis are considered to be structural designers and their networks.

With this in mind, Hybs and Gero (1992) studied evolutionary design processes and concluded that "The environment should be understood in the holistic manner as one whole including the product or artifact itself. The design process is then seen as an integral part of the evolution of the total environment and its progression towards a new state." From this quote one may infer that the artifact is not the only item in the design process with a seed and an evolutionary process. There are at least three types of seeds and their growth in relation to the holistic design process. Such seed

[21] In the words of Milliken, they "Steal Ideas Shamelessly" (fertilisation).

types correspond to the earlier mentioned forms of Leviathans (Figure 2.6):

1. Artifact growth (SDP):
   Much of the standardised, adaptable SDP design knowledge is contained in DNA. The structural designers form the RNA who decode and regulate this DNA. That instantiated design[22] result in actual, produced parts of the artifact (proteins). These are transported and put in place (structuralism), and eventually the artifact grows into a mature organism. It becomes part of a community (e.g. a project with other artifacts) and if it proves successful within that context, it may produce (slightly deviated) offspring. The process of natural selection may be sped up by applying continual breeding, genetic manipulation or even gene therapy (designers do not always wait for natural selection to happen). Finally an artifact arrives at the end of its natural life, decomposes and ideally nourishes new artifacts (reuse). Design team growth and design growth follow similar processes but at different scales.

2. Structural design team growth (tacit SDP knowledge):
   A structural design team's quality and learning capabilities over a long time-span dictate the artifact growth. Since designers are the artifact's RNA, improving the design team implies improving their ability to decode artifact knowledge (DNA). The combined DNA-RNA system will then result in better artifacts. Note that RNA, proteins, body parts etc. are 'embodied levels' with capabilities that do not have to be encoded explicitly in DNA.

3. Design growth (Design Science):
   Design growth is the attempt to how designers design. Again, this is an evolutionary process but at a very high level.

2.3.2 Leviathan's DNA

Various authors have explored DNA in the context of AEC products. Examples come from AEC enterprises with names like "Building DNA" and

[22] An instantiated design is not equal to models and blueprints but constitutes of various pieces of specific, contextual design at the right place and time.

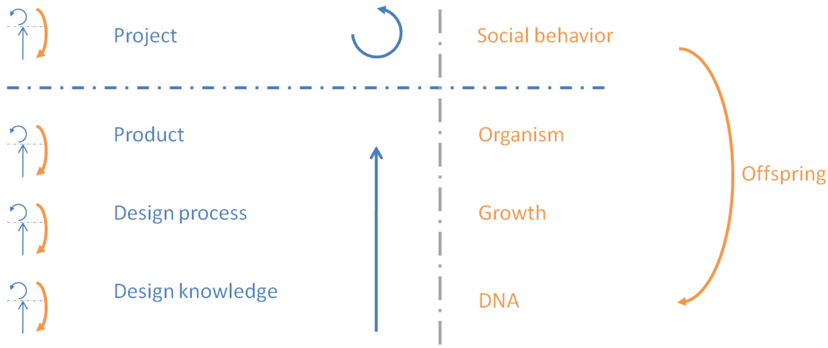


Figure 2.6: *Design knowledge is coded in DNA. With the help of designers (RNA) in a design process, it grows into an organism (instantiated design, product), exposes social behaviour with other products in a project and finally produces offspring. A similar discussion is valid for the distinct items. For example, design knowledge itself grows with the help of researchers, interacts with other knowledge areas and produces offspring if it proves successful.*

"BIM-i DNA". They often use DNA synonymously for BIM. However, BIM describes a structural design product in a very direct, explicit and project-specific form (geometry, planning, risks etc.), and hardly touches the implicit synthesis process. Therefore BIM describes Leviathan's instantiation (phenotype, final SDP form), not its genotype (i.e. compressed SDP knowledge). The know-how to produce the phenotype comes from DNA together with the structural design team (RNA, Section 2.2.4). A possible definition for Leviathan's DNA therefore is (also refer to Fischer et al. (1994)):

> Leviathan's DNA is the interrelated, compressed and adaptive knowledge that - if put in the proper context - defines Leviathan's growth and maintenance. Leviathan's RNA constitutes of the team members who query the DNA (Figure 2.7).

Such definitions will result in Leviathan DNA that includes structural design sources (websites, libraries, other colleagues, design codes, handbooks, experience etc.) that structural designers use to produce their SDP. This SDP DNA must be adaptive to be of competitive advantage. In addition, RNA and DNA must be in harmony; they cannot operate separately or be mixed with RNA and DNA molecules in other SDP; RNA must know

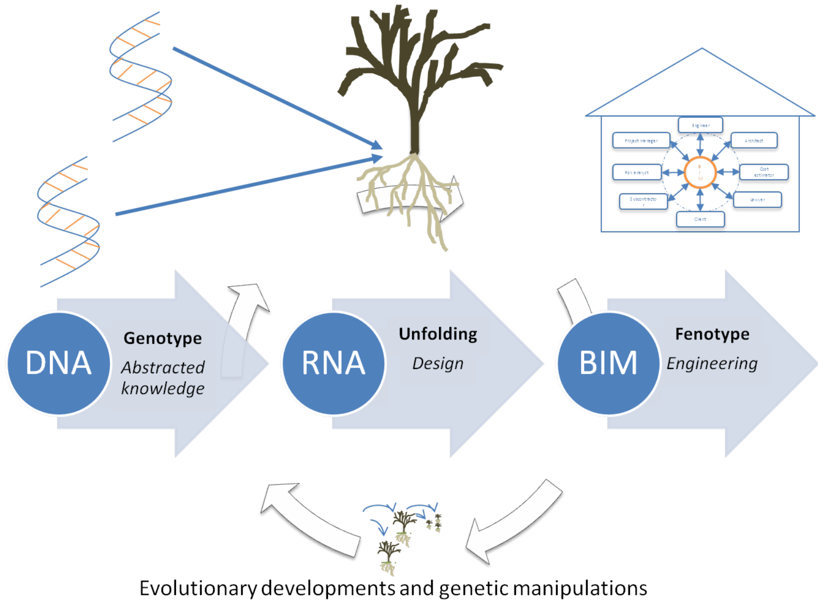


Figure 2.7: *BIM is the phenotype since it is explicit, project-specific and related to the artifact form. To arrive at the phenotype, it requires RNA (a design team) to translate abstract structural design product knowledge (DNA) into constructable, context-specific solutions. Note that there is no single, centralised DNA repository. Even though DNA is an abstraction of instances over many years, DNA itself is also an instance.*

exactly how to unfold DNA in order to arrive at the necessary knowledge[23] - "...knowledge is inseparable from thinking and acting" (Fahey and Prusak (1998)).

Using Leviathan DNA in this way implies that knowledge will be stored in a very redundant, distributed and designer-specific way - much in the same way it currently happens. Nature for example does not pull DNA from a centralised database for each new organism; rather it depends on the 'repositories' of the individual organisms and their meiosis.

This partially eliminates the need for maintaining structural design object libraries manually. Libraries for anything more than very standard, static SDP quickly become outdated since only a small number of people

[23] Nonaka and Takeuchi (1995) described how successful Japanese designers use a similar model. They highly value tacit knowledge and trial-and-error experiences (they are affiliated to their designs), while western countries emphasise training and explicit, transportable knowledge.

have the ambition to maintain them properly. On the contrary, within an evolutionary environment, successful SDP will bubble up and allow for quality managers to certify them rather than inventing, maintaining and imposing them. Libraries should therefore be composed of *time-lagged instance*s, not of *classes.* That is what similar to how nature recombines individuals' DNA, preserves it for a product lifetime and finally passes a slightly adapted library on to a next generation[24].

It may be concluded that evolutionary processes naturally combine adaptability with standardisation. Structural designers may benefit from this type of standardisation, provided that the evolutionary environment to do so is available.

2.3.3 Allowing for the evolutionary process to happen

In order to allow for SDP's evolutionary process to happen, structural designers should have extremely easy access to the environment, lavishly filled with historical SDP in order to i) copy them, ii) improve them and to iii) release their modified or new SDP (Figure 2.8). Social media and merchandising sites are very good at this process. The mindset in structural design should be to apply similar techniques en mass to SDP.

Whenever SDP are copied, references must be maintained between the new copies and their ancestor(s). That allows to trace the example's offspring over several generations and to visualise its evolution. Obviously a copied example may be composed of several good examples (ancestors). An n:n relationship is sufficient to model both the ancestors and their offspring.

Any one SDP should follow its own evolutionary process, whether it is small or large. It is tempting to treat a complex composition as simple Lego blocks that have been piled to something very complex. Complex products cannot be composed of standard low level modules in a generic plug-and-play way since all modules are continuously subject to modifications and competition. In addition, it is at least very difficult to compose a large variety of beautiful, complex, custom products with the bare minimum of building blocks. In a Lego example, specials like a helicopter or a motor cycle require custom Lego pieces (e.g. a curved window or a helmet).

[24] This process may be very diffuse, since knowledge managers may manipulate or prescribe certain solutions or even dictate the selection mechanism.

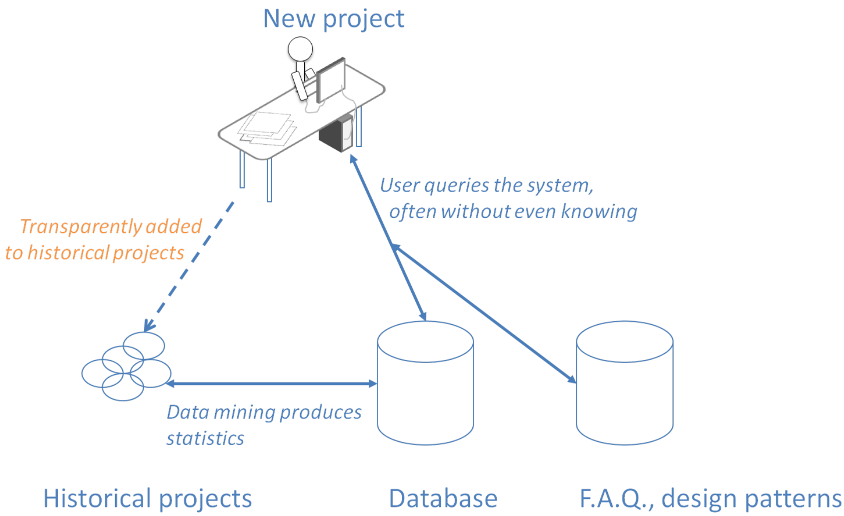


Figure 2.8: *The structural designer queries and fills the system in symbiosis; by doing so he copies and adapts SDP. These small mutations contribute to a large variety of similar SDP, and after some time it will be possible to apply SDP without the need to create additional calculations.*

Structural designers will find more than only SDP in a CBR system. The associated context, related projects, best practices, calculations, design codes, decisions and such pop up together with the example and therefore the SDP has similarities with a microcosm. A microcosm is a complete miniature of something larger (which is unrealistic to apply in everyday life). An epitome is a biased, compressed version of it for a specific purpose (Figure 2.9). DNA but also the human mind are profound examples of epitomes (Appendix C).

2.4 Null hypotheses

The elements of the theory as described earlier in this chapter may be verified by formulating and testing null hypotheses (as mentioned in Section 1.4). It is beyond the scope of this thesis to test all the null hypotheses that follow from the theory.

- Collective mind

    1. Chaos and complexity in structural design environments do not

Figure 2.9: *An example of an epitome is a person sitting in a building who thinks about the building and its environments. It is a combination of both decomposition (person in the building) and something bigger inside the decomposition (the thought about the building inside the person's head).*

contribute to waste reduction.

2. An extremely easy to use HMI does not yield to more SDP.

- Seed centred design

  1. Structural designers do not share SDP.
  2. Structural designers do not use other structural designers' SDP.
  3. Structural designers use copied SDP unchanged.
  4. The majority of SDP is one-of-a-kind.
  5. Structural designers remain independent of their own SDP.
  6. The structural design environment itself is not subject to evolution.
  7. Libraries do not emerge from SDP.

The prototype (Section 3) is the instrument for falsifying the null hypotheses, and the results will be presented in Section 4.4.

CHAPTER 3

Prototype

"You can buy a chess machine that beats a master but you can't yet buy a vision machine that beats a toddler's vision" - Donald Hoffman

During this research, the author created several prototypes (Figure 3.1) to demonstrate certain aspects of Structural Design Product (SDP) evolution, the corporate mind and the Leviathan in context of structural design. These prototypes were developed at a preliminary stage (Section A). The FrameDesign / CloudConstruct prototype resulted in the largest amount of measurable data and therefore the author elaborated on it.

3.1 Prototype design

The author examined the available platforms on which a prototype could be developed, such as web-based applications, smartphones and desktops. The smartphone as a platform was chosen since it allowed to produce and distribute a product rapidly, it was widely available, and it allowed for quick feedback. The main prototype is a family of apps for Android smartphones (Figure 3.2), which consists of

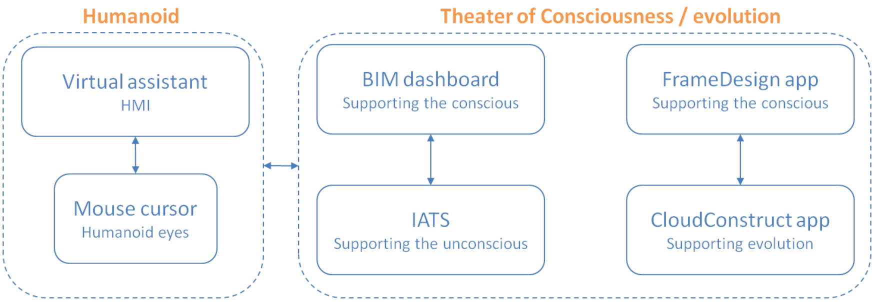


Figure 3.1: *These prototypes were built during the course of this research. A number of them were built to experiment with the Theatre of Consciousness and evolution, while the virtual assistant together with its associated visual systems represent a Humanoid.*

- FrameDesign: a 2D Finite Element Method (FEM) implementation.
- CloudConstruct: tool to share FrameDesign structures with others.
- Various other apps (Engineering Libraries, Concrete Design, Channel Design, Weld Design, Parallelogram, Circle of Mohr, also produced by the author). Those proved to be less useful for this research, mainly due to the lower number of active users compared to FrameDesign.

FrameDesign offers much of what a user would expect from a simple 2D FEM application: Graphical input and output, beams, nodes, various types of forces and supports, profiles, load cases and load combinations, results (internal forces, stresses and displacements), document output, metric & US units, export/import and more. This app belongs to the conscious since the structural designers can oversee what they design. For the experiment to be successful, it was important to have a large user base. Therefore - even though FrameDesign was not the main goal of the prototype - much effort was put into creating a user friendly and functional app[1] (Figure 3.3). The reason for choosing to develop FrameDesign as a vehicle for demonstrating evolutionary development, is that frame structures are often one-of-a-kind SDP that are difficult to standardise.

CloudConstruct is complementary to FrameDesign. It is intended to serve as the unconscious and to provide the evolutionary environment in

[1] The first version was a simple 1D beam design app, but it was upgraded to 2D to attract more users.

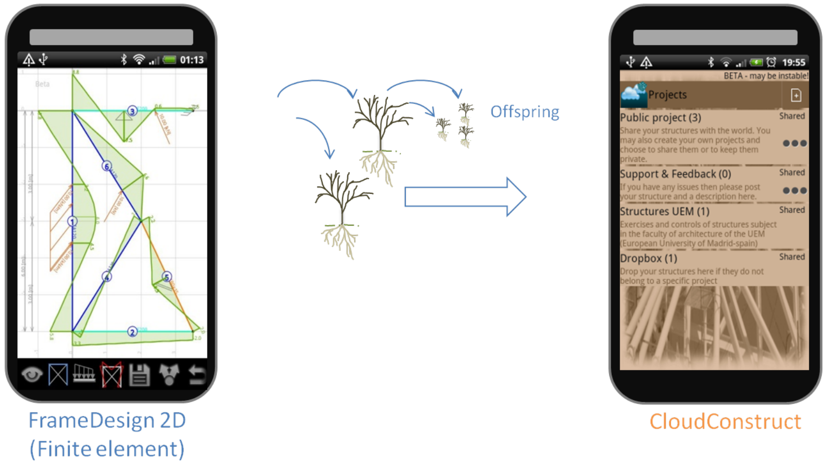


Figure 3.2: *The FrameDesign / CloudConstruct prototype was intended to collect data about SDP evolution. An engineer creates a frame structure and shares it by using CloudConstruct, which keeps track of the specific structure's offspring. That provides information about a structure's evolution.*

which SDP evolve. Basically, it allows users to save and share their FrameDesign structures in the cloud and to let fellow structural designers share and copy structures. Structural designers can organise their uploaded structures in projects. They have the option to make a project public or private. Structural designers may copy structures by:

- selecting a shared public structure within CloudConstruct:
  After selecting a structure in CloudConstruct, it is downloaded to the user's device and opened with FrameDesign for further analysis and mutations.

- scanning a QR code:
  A user may generate a QR code for an active structure and let a fellow structural designer scan it, after which - through the intermediation of CloudConstruct - a copy will appear in latter's FrameDesign.

After creating a copy (offspring) in either ways, it is saved to the receiver's own CloudConstruct project together with a link to the ancestor from which it was copied. The newly copied structure may be shared again, after which CloudConstruct maintains the new offspring relations as well. This res-

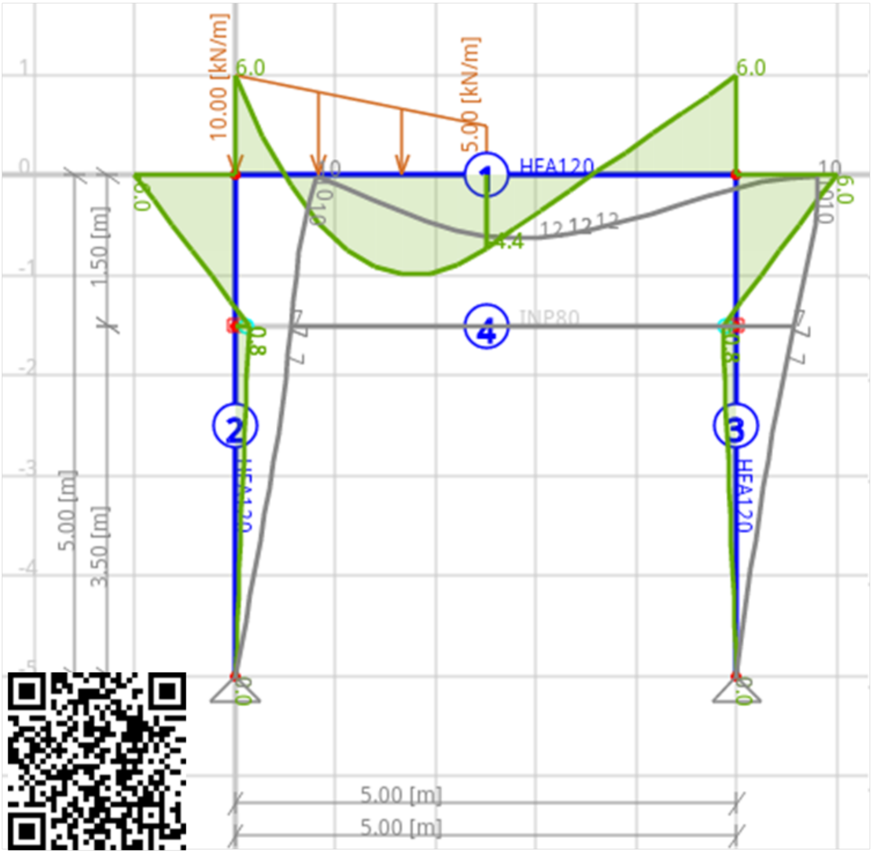


Figure 3.3: *Screenshot of the FrameDesign Android application.*

ults in an hierarchical tree in which the evolutionary development of frame structures becomes visible.

3.2 Data collection

The author collected data using a variety of sources: Android Market statistics, YouTube statistics, AdMob statistics, Google App Engine, email, and a questionnaire. Google App Engine and the questionnaire provided information about the amount of evolution and unconscious design that happened, while the other sources provided information about the environment that enabled it. These techniques and the results are briefly described in the following paragraphs.

3.2.1 Sampling source

Although the author is unaware of the amount of structural designers owning an Android device, it may be assumed that there is no selection bias if structural designers with Android devices are used as a sampling source. Android devices can be assumed to be sufficiently used by the majority of professions and countries. As an indication there were ~850k device activations/day and ~1 billion app downloads per month in 2012. According to Gartner (May 2012), Android's 36% market share was the largest of all mobile operating systems.

FrameDesign was made available free of charge and with unrestricted features to ensure that the entire sampling source could be reached. The same is true for CloudConstruct (although users had to agree to provide their account information).

There were no quality controls to ensure that the correct users were reached; however, the quality could be measured afterwards with CloudConstruct: unusable structures would simply remain unused by others. Therefore the amount of offspring generated in CloudConstruct is a quality measurement. This is not entirely true, since very specific products will get very little offspring too but receive better comments and ratings. Within this prototype, comments and ratings were not yet implemented. As an alternative, the author attempted to measure what type of users (students, professionals, just playing) were using the prototype by using a questionnaire.

Assuming that the samples in the following sections are simple random samples from a large population, the indicative margin of error the at a 95% level of confidence[2] can be calculated with

$$\frac{0.98}{\sqrt{sample\ size}} \tag{3.1}$$

3.2.2 Android Market statistics

The Android market tracks the number of downloads, active installs, types of devices, demography, comments, ratings, other apps that users installed

[2] That is, if this survey were repeated 100 times, then 95 times one may expect to find the given values within the margin of error. A higher level of confidence gives a higher margin of error.

and such. FrameDesign reached ~160k downloads of which ~50k were active installs. FrameDesign and its associated apps appeared on many third-party websites that collect useful software applications for structural design, and it was listed in Android's top 300 productivity tools.

Although there are far more popular apps (e.g. Autodesk's AutoCAD WS with 1 million+ downloads), the author assumed that the FrameDesign user-base was sufficiently representative for this research. A proof for this assumption is that Autodesk's app ForceEffect (released very recently, with similar functions to FrameDesign) did not exceed 10k-50k downloads within the first few months after its release even though Autodesk is a "top developer" that lifts on its other apps' successes. The author also unpublished FrameDesign and re-released it as a new app (with the same name) to check how fast users would switch; those users could be regarded as genuinely active users. Within less than two months (during the summer holidays), it reached the 10k-50k range.

The stable growth rate of the active installs indicates that users are loyal to the app (Figure 3.4). Many students and professionals reported to use the FrameDesign / CloudConstruct system. They requested features, contributed translations in 10 languages and pointed out numerous bugs.

This is precisely how the intended evolutionary SDP development should be[3]. One person proposes a product, others criticise it or contribute to it. However, the product in this case was mostly FrameDesign itself, while the author attempted to allow for evolutionary SDP development. Therefore, CloudConstruct should be positioned like an "AEC SDP market" app with possibilities such as commenting and starring SDP, much like the Android market itself. It could also serve as the CBR engine that provides users with statistical feedback on historical designs.

Without anticipating it, the author became very engaged in improving FrameDesign - even apologised to angry users (who regarded FrameDesign as 'their' app) for changing things - and therefore FrameDesign with its associated users became a Leviathan in which the author became the RNA.

[3] This model is very similar to Open Source developments

(a) *Active user installs for FrameDesign since December 2010 reached ~50k in 2012 (~160k total downloads). Point 1 was a mistake in the user count, which Google corrected later. Point 2 is more interesting; at that point, the Android Market associated FrameDesign with the popular AutoCAD WS. Users who installed AutoCAD WS received the suggestion to install FrameDesign as well, which boosted the install numbers. Note that the active installs growth rate is very stable - almost a straight line. It does not follow a so-called 'hype curve'. Apparently the users are very loyal to the app.*

(b) *FrameDesign was updated ~250 times in response to bug fixes, feature requests etc. This Figure shows how responsive people were to new updates. Within ~10 days after an update, 70% of all users installed the new version; even in case of frequent updates. Updates that were released within less than 10 days after a previous release, resulted in user complaints.*


roy on Thursday, May 31, 2012 at 21:38 | Sony Ericsson Xperia X10 Mini Pro (U20i) | Version 2.84 | Spanish

Muy. Buena app me sirvie bastante

Sergio on Tuesday, May 29, 2012 at 07:13 | Samsung Galaxy Ace (GT-S5830L) | Version 2.84 | Spanish

. Excelente

mathumphreys on Sunday, May 27, 2012 at 15:22 | Samsung Galaxy Note (GT-N7000) | Version 2.79

Fantastic! Love this app, dxf line import gets it 5 stars. I use this on my galaxy note and acer iconia a500. Id like to see the controls change from the current icons to the more common pulldown menus. Particularly on ics.

maxence on Saturday, May 26, 2012 at 18:13 | galaxys2 | Version 2.83 | French

Super Comme dit ci dessous, super pour faire de petits exercices de meca. A voir pour un usage plus pro. Même si je n'ai pas encore détecté de bugs, j'ai du mal a avoir confiance. Ah oui, 3euros ... c'est dérisoire pour un outil de cette qualité (Étudiant en genie civil)

(c) *Ratings and comments (866 in total) were an important drive to keep improving FrameDesign. Therefore it may be assumed that if structural designers could publish their own SDP on an "AEC SDP Market" equivalent that integrates into their daily tools, they will have that drive as well - especially if they can generate revenue or gain a respectable position with it.*

Figure 3.4: *FrameDesign statistics. [source: Android Market Developer site, 2012]*

3.2.3 YouTube statistics

A valuable source of information comes from a promotional video about FrameDesign on YouTube. Since YouTube tracks gender, location, age, device, browser and such, one can analyse the type of people who watch this video. According to these statistics (Figure 3.5), middle-aged men in the United States is the primary group. That might indicate that the majority of the FrameDesign users are professionals.

3.2.4 AdMob statistics

AdMob is an advertising network that can be integrated in apps. AdMob's goal is to monetise apps, but it provided information about the frequency with which FrameDesign was used: at least ~43 hours/day (Figure 3.6). Users without an internet connection could not be monitored with AdMob.

3.2.5 Google AppEngine

Google AppEngine is the back-end cloud system, exposed by CloudConstruct to which users published their structures. It provides a reliable high-replication no-sql[4] database and interfaces that easily integrate with Android devices and websites. Using CloudConstruct involved three steps:

1. Registering the user's device with AppEngine to identify the active user (using Google federated login)

2. Creating a project

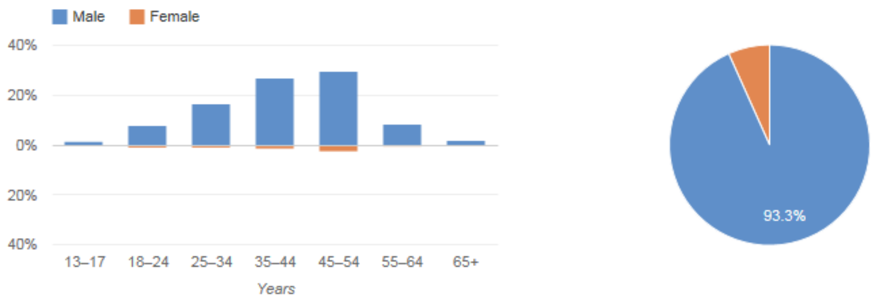
3. Uploading structures into that project

From the acquired data in AppEngine, the author extracted that to date,

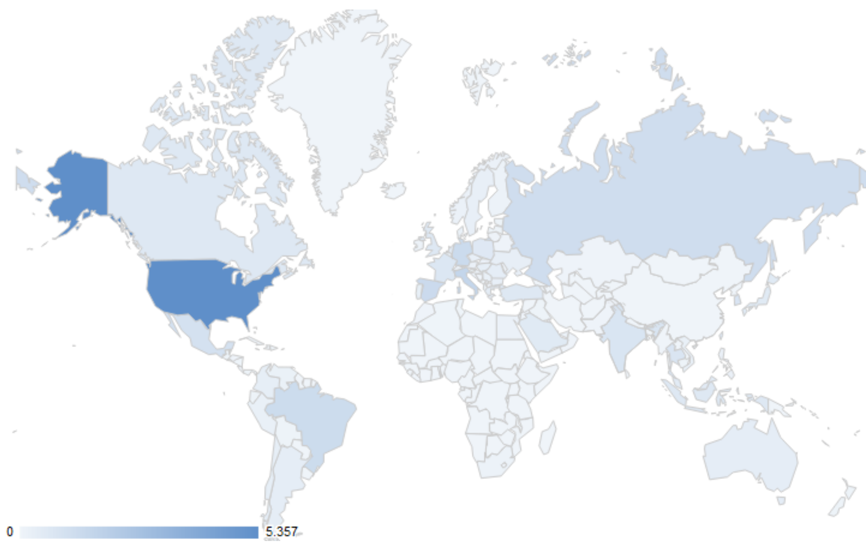
- 1732 users registered their devices with CloudConstruct. The total amount of CloudConstruct downloads was 2761, so 62% of these users were able / agreed to provide their identity to the system.

- 697 projects were created. It seems as though it was not evident for the users that they were to continue after this step and to upload

[4] A no-sql (not only sql) database is schema-less, and allows for large amounts of hybrid data to be saved and retrieved.



(a) *Age and gender*



(b) *Demography*

Figure 3.5: *The majority of the people who viewed the You-Tube video that was associated with FrameDesign are middle-aged men in the United states (based on a total amount of ~30.000 views). Although it cannot be proven from this data, it does indicate that these are professional users. [source: YouTube personal video statistics, 2012]*

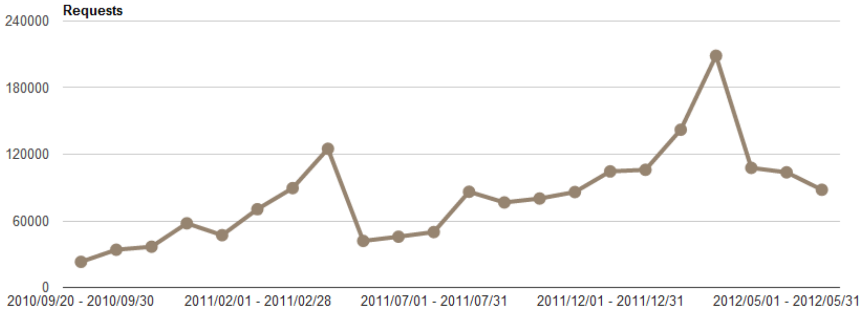


Figure 3.6: *AdMob showed that there were ~80.000 ad requests/month. The ad refresh rate was 1/min, therefore the users collectively used the app ~43 hours/day. Users without an internet connection could not be monitored, and therefore the actual use was higher. [source: Admob developer site, 2012]*

the actual structure as well. Therefore more projects than structures existed in the database, while the ratio should be the opposite. 85 of these projects (12%) were public.

- 553 structures were successfully uploaded by 335 unique users. Users were able to create structures in two ways:
  - By explicitly opening CloudConstruct, creating or selecting a project and finally saving the structure within that project. Assuming that these users provide structures with meaningful names, the author concludes that only ~50 structures (10%) were created in this way.
  - By pressing the auto-upload button or using the QR copy function. The QR copy function generated a QR code for a specific structure so that a colleague could scan that code and get a copy of the structure. These convenient upload methods generated 90% of the uploaded structures.
  - A future, arguably more successful method would be to keep the projects and structures on the user's SD Card in sync with CloudConstruct. Saving structures to the SD Card is easier for users since it is fast and offline. Keeping these structures in sync with CloudConstruct would enable them to transparently backup their data, while at the same time it would provide more

and higher quality products in CloudConstruct without any additional effort.

23% (error margin $\frac{0.98}{\sqrt{553}} = 4\%$) of the uploaded structures were copies (offspring). Some of these copies remained unchanged (clones), some slightly changed (mutations) and others were modified severely (genetic manipulations). Examples are shown in Figure 3.7.

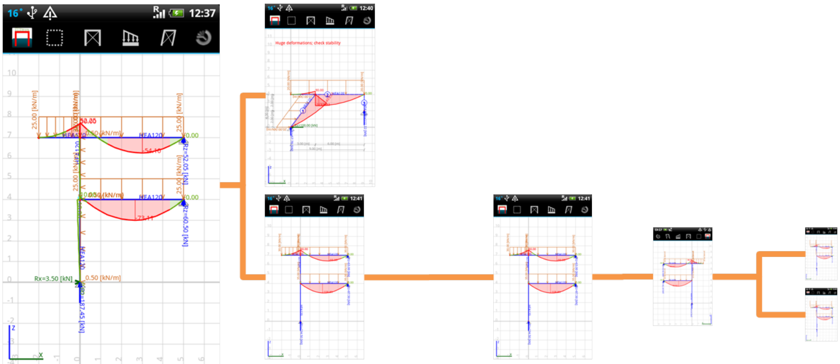
3.2.6 Questionnaire

Within FrameDesign, a prominent link to a questionnaire with six simple questions was included. It was available for a few months, and resulted in 31 responses (Figure 3.8). Due to the low number of responses, the indicative margin of error is considerable: $\frac{0.98}{\sqrt{31}} = 18\%$. The most important information in the questionnaire is that

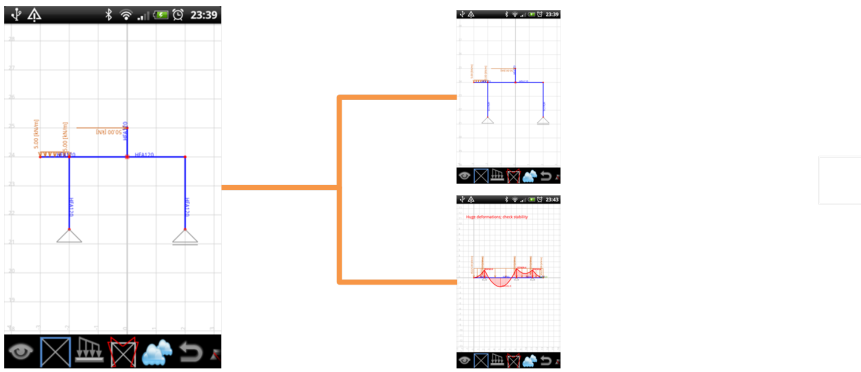
- Professionals and students are the main FrameDesign users
- Most people use FrameDesign weekly / daily
- The general impression (interface, features, responsiveness, support, help and sharing) is 'good' (the average rating on the Android market is 92%);
- ~88% would like to see more evolving products in CloudConstruct (not only frames).
- ~18% use CloudConstruct to publish their structures or to evolve existing ones.

3.2.7 Relation with other apps in the Android Market

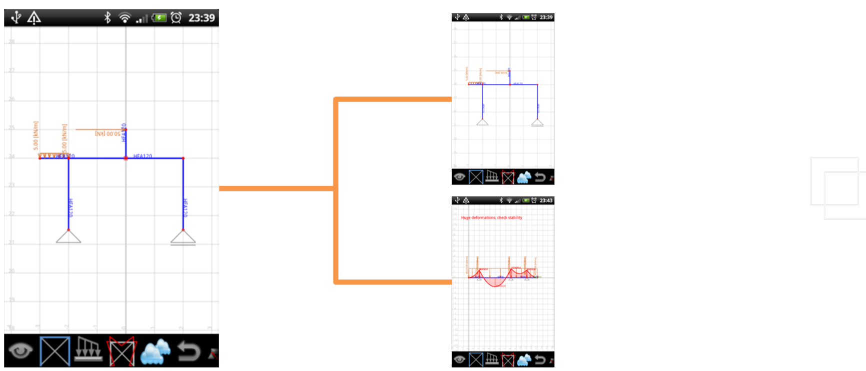
The android market hosts many engineering related apps. It regarded FrameDesign, AutoCAD WS and AndCAD (a 2D truss application, one of the author's competitors) as related apps. As soon as a new version of the popular AutoCAD WS was released, both FrameDesign and AndCAD followed a similar 'hype' curve as AutoCAD WS as shown in Figure 3.9, even though FrameDesign and AndCAD did not release a new version in that period. The author expects that designers' products in a market-like



(a) *This structure was branched by a single person. There was one severe change, the rest of the structures were small variations up to 5 levels.*



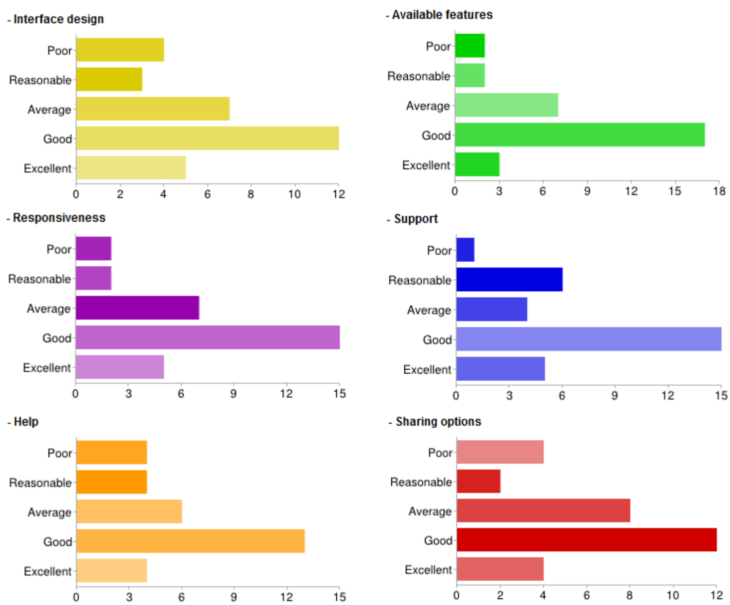
(b) *One severe change, several exact copies by multiple people, only one level deep.*



(c) *One severe change, many exact copies and slight variations by multiple people, but only one level deep. The parent structure was uploaded by a teacher in Greece, his students were asked to download the example and to answer several questions.*

Figure 3.7: *Examples of FrameDesign structures, as exposed by CloudConstruct. Some of the copied structures remained identical, some were small variations (mutations) and some were radically different (genetic manipulations). CloudConstruct keeps track of these multiple generations.*

(a) *Usage frequency and professional use*



(b) *Ratings for various aspects*

I publish my structures in the cloud using CloudConstruct	**4**	24%
I use other people's structures in CloudConstruct	**3**	18%
I find it usefull to start from existing structures rather than starting from scratch	**3**	18%
It is interesting to see how other people use my public structures in their own projects (using the Offspring button)	**2**	12%
I would like to see more products in Cloudconstruct (not only structures, but also bolts, profiles and such)	**15**	88%

People may select more than one checkbox, so percentages may add up to more than 100%.

(c) *Sharing options*

Figure 3.8: *Questionnaire results. [source: Google docs response summary, 2012]*

Figure 3.9: *One product may boost another. In this case, the Android market regarded FrameDesign, AutoCAD WS and AndCAD as related apps. As soon as a new version of the popular AutoCAD WS was released, both FrameDesign and AndCAD followed a similar hype curve as AutoCAD WS. [source: Android Market statistics, 2012*

environment will follow similar coupled patterns and may show forms of co-evolution.

3.3 Conclusion on prototype

Over time, a hierarchical tree structure that represents the copied structures became visible within CloudConstruct. The majority was copied only one or two levels deep, since copied structures immediately become private unless the user explicitly shares them. A number of structures were copied up to 7 levels, during which multiple users were involved. The shared structures with appealing names (such as "control 8 ejercicio 1" or "beautiful deflection") produced reasonable offspring (10 or more siblings) while structures with default names (e.g. "New project") produced very little or no offspring. It may be expected that more copying can be achieved by:

- improving the CloudConstruct interface and the back-end cloud system
- allowing automatic synchronisation between the saved structures on the local SD Cards and CloudConstruct
- allowing users to be notified on changes in projects of their interest

- including thumbnails (attractive instances are more likely to produce offspring)

It is possible to extract useful information from an ever increasing number of evolving structures, including:

- listing the most popular steel profiles, for example HEA or Wideflange sections. A number of custom profiles that users created themselves were uploaded together with the structures. It becomes possible to generate libraries instead of maintaining them.

- building phylogenetic trees. Instead of relying on the hierarchical trees (as shown in Figure 3.7), it is better to identify similar structures throughout CloudConstruct. E.g. after a genetic mutation, the type of structure may fit better with another family of structures than the hierarchical tree it was copied from.

- identifying experts. The authors of popular structures may be identified as experts. The identification process may be enhanced by introducing stars, comments and other social media features.

- It could be possible to generate eLearning material or simple games in the form of optimisation problems, which could help the person who creates the game to find his optimal structure with the help of others. A first (implemented) simple step in this direction is that users can now insert text and parametric formulas (much like MathCad) in FrameDesign.

An unconscious feature the author started to implement (using the OpenCV library for Android) is to capture images from a phone's camera, convert it to a line drawing (through a Hough Transformation) and match those lines with a best fit in CloudConstruct (CBR). First it was intended to let the user sketch a structure on paper, take a photo and transform it into a FrameDesign structure. However it proved highly complicated due to 1) the large amount of lines/noise in a photo, and 2) to decide on how and where elements should be connected. As a first best guess, the multitude of uploaded structures in CloudConstruct could serve as a case base. CloudConstruct may be able to find a best match, which includes relevant load cases, load combinations and project context.

CloudConstruct is only a prototype that is still far from perfect. The complexity of client-server communication, federated login, synchronisation and other technical issues could not be fully resolved during this research. A similar but flawlessly functioning tool will arguably produce better results.

CHAPTER 4

Conclusions

"Intelligence is what you use when you don't know what to do" - Jean Piaget

THIS chapter provides the conclusions on the theoretical framework (Section 2) and the prototype (Section 3). It answers the research questions and tests the null hypotheses. After depicting the limitations, advantages and effects on the AEC industry, future research will be encouraged.

4.1 Main conclusions

During this research, a theoretical framework was developed and tested with prototypes in several iterations. This section provides the main conclusions.

4.1.1 Theoretical framework

The AEC industry is a complex, distributed and chaotic system in which virtual enterprises conduct both repetitive and one-of-a-kind projects under the pressure of heavy competition. There is waste due to the sub-optimal organisation, but the industry continuously adapts to a changing world and keeps producing useful buildings and infrastructure. Therefore, the AEC industry in general and structural design in particular has the characteristics of a Complex Adaptive System (CAS).

Many initiatives - inspired by mass production industries - aim at changing this complex organisation into a more centralised and regulated organisation in order to reduce human errors, to improve communication and to use computerised systems more efficiently. However, after decades of developments, it proved difficult to impose such unifications on structural design since

- design problems are wicked (Footnote 1 on page 14) and therefore they cannot be solved in a predictable way (the process of synthesis is chaotic)

- imposed mass production approaches are subject to variations and competition, and therefore result in new spaghetti models (Figure 2.2)

Structural design has evolved into a CAS in order to address wicked problems with the help of human "neural networks". If it is accepted that structural design is a CAS, then mass production principles and chaotic complexity are but a paradox. A useful metaphor is to compare structural design with nature, with the key elements being:

- Natural selection:
  This useful mechanism does not happen to a sufficient extent in the AEC industry. Designers should have extremely easy access to a lavishly filled source of relevant historical solutions[1] rather than having to start from scratch, attempting to create standardised ecumenical design models or being limited to small sources (continual breeding). An "AEC SDP market" will stimulate standardisation, diversity and finding relevant expertise.

- Collective mind:
  Swarms, flocks and other network-based social communities are successful at solving problems collectively. There is unused potential in using the Internet to solve structural design problems collectively (i.e.

[1] That is, stimulating to Steal Ideas Shamelessly (Miliken) to propel a learning organisation (Hübner (1995)).

in a Leviathan environment). Techniques such as CBR may use statistics to assist in finding known or heuristic solutions to problems if human interference is unavailable.

- DNA:
  Nature uses a redundant, distributed knowledge base (DNA) rather than centralised sources. A product's "DNA" constitutes of sources of information that designers use to instantiate a product's design into an actual, context-specific design. In this view, designers are the "RNA" who decode the DNA information.

Tools that are extremely simple to use could arguably improve the design process[2]. Therefore a first simple implementation of a suitable design support system for structural design is a basal set of well-performing extremely-easy-to-use tools (e.g. in-place and with natural humanoid interfaces) that allow designers to evolve SDP (e.g. a market) and to use the corporate mind efficiently (e.g. CBR).

The result is a win-win situation; structural designers will provide the Leviathan (the design support system) with useful information (from which usable statistics and libraries can be derived) in exchange for access to collective SDP knowledge (Figure 4.1). If structural designers do not properly implement such a Leviathan design support system, then they are likely to remain viruses (Footnote 20 on page 28) or they will lose colleagues to other (perhaps freely available) Leviathans, which may cause a severe loss of unconscious power.

Leviathan SDP support systems may be equipped with humanoid interaction and referred to as virtual assistants[3]. Virtual assistants may conglomerate into new enterprises and even expose their own expertise. They may relieve structural designers from being computer Leviathan RNA (that is, that structural designers must conform to computer interpretable formats in order to perform structural design tasks).

There is a multitude of SDP that can potentially evolve, however the required evolutionary environment is missing. Therefore new SDP continuously appear, but immediately disappear without any offspring. Therefore

[2] Wicked problems require wicked tools.

[3] Virtual assistants are currently visible on smartphones; people talk to their phones to find answers to natural language questions. These virtual assistants use popular search engines or knowledge systems (e.g. Wolfram Alfa) to find the answers.

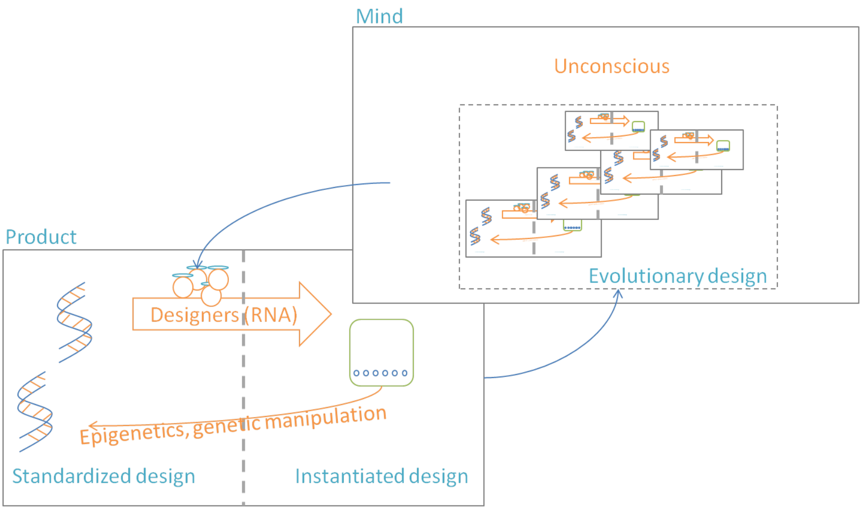


Figure 4.1: *The socio-evolutionary framework. SDP are subject to evolutionary product development, and allow to form the collective unconscious (CBR). Structural designers may use CBR to validate new instantiated designs against historical equivalents.*

the first step should be to allow for SDP evolution to happen, for example by ensuring that structural designers can extremely easy:

- access a lavishly filled live product database (population)
- determine an existing SDP fitness for new projects (natural selection, e.g. by ratings or cost)
- copy an existing SDP (produce offspring)
- adapt SDP (mutations, genetic manipulations and epigenetics)
- publish adapted or new SDP
- attach information to existing SDP (e.g. best practices, comments, ratings, etc)

Certain SDP may be successful while others disappear (perhaps together with their RNA designers). Structural designers will find a wealth of examples to build upon, whether they are simple SDP (beam) or very complex (a fly-over). The need to perform a full calculation for new SDP in the early stages of structural design reduces since the dataset will in time contain many variations of SDP and therefore allow for heuristic-like CBR.

4.1.2 Prototypes

The author created various prototypes to demonstrate a possible implementation of the basal framework and its practical use (Section 3.1 and Appendix A.1). Although the prototypes were very limited, the results support the theoretical framework. For instance, the FrameDesign / CloudConstruct prototype was downloaded 160k+ times and resulted in 553 structures and offspring. Users were prepared to provide their identity (email address) to a Leviathan in the prototype (CloudConstruct) in exchange for an exchange platform with evolving SDP.

A prerequisite for attracting enough users to the prototype was that the prototype and the infrastructure worked flawlessly and intuitively. The author had to continuously respond to users' comments and to gradually improve the app in order to keep attracting new users. The tools and infrastructure themselves should therefore be regarded as socio-evolving products as well. During the process, the author became the RNA within the app ecosystem.

The FrameDesign / CloudConstruct prototpye has shown that ~23% (+/-4%) of the structures that designers uploaded, were evolving (i.e., produced offspring were subject to mutations and genetic manipulations). The prototype's user friendliness greatly affected the outcome. Moreover, not only the SDP within CloudConstruct evolve, FrameDesign / CloudConstruct were subject to an evolving ecosystem as well (the Android market) in which the author actively competed with other similar apps that appeared. Synthesising both observations the author concludes that if designers are offered a Socio-Evolutionary AEC Market (SEAM) similar to the Android market, that integrates well with their daily work and allows them to publish and to download relevant (paid) SDP, they will be actively engaged in improving those SDP. If SEAM additionally allows for CBR-like calculations, then collective SDP knowledge may be mobilised en mass.

FrameDesign / CloudConstruct is only a simple proof of concept in this direction, but the author expects that a sufficiently large and well performing SEAM will lead to waste reduction.

4.2 Added value of this research

The added value of this research is that it may provide a comprehensive reference framework for gradually improving structural design support, without revolutionising the current system. If the nuances of a Socio-Evolutionary Complex Adaptive System are taken into account, then it may allow enterprises to build structural design support systems in which conflicting or one-sided approaches to waste reduction may be harmonised.

4.3 Answers to the Research questions

- *Question 1:*
  In what way could the advantages of mass production principles be combined with decentralisation and non-uniformity in order to effectuate waste reduction in structural design?
  Answer:
  Nature combines adaptability with standardisation. Similarly structural designers' access to a wide range of relevant, up-tot-date and perfected SDP that have been tested in a large, evolutionary environment will reduce the need to repeat SDP design actions, and therefore contributes to waste reduction.

- *Question 2:*
  How could knowledge management be integrated in structural designers' daily work effectively and without disproportional effort?
  Answer:
  If there is a large amount of evolving SDP, then CBR may be used to assist structural designers both in finding relevant products and in judging their SDP offspring. The associated socio-evolutionary system can be materialised with:

  1. Historical SDP:
     Structural designers must have extremely easy access to historical SDP, which will allow SDP evolution to happen.
  2. Live SDP knowledge management:
     New designs must be incorporated in the historical SDP base in-place and with minimum effort. Regularly used SDP form a live

evolving library.

3. Reward system:
   Popular SDP will be rated and commented positively, and much additional data such as photos and best practices will be attached to them. In addition, the respective structural designers will build a certain reputation, and they may generate revenue from their SDP. High rewards may be an indication to knowledge managers for deciding on SDP certification.

4. Leviathan:
   The unconscious is the implicit SDP information that can be accessed through CBR and live genetic algorithms. A well-oiled Leviathan socio-evolutionary system may become a virtual assistant (self aware) with natural humanoid interfaces and a perceptual world similar to structural designers. It may form alliances with structural designers and other Leviathans.

4.4 Null hypotheses

The prototype in Section 3 was designed to test the null hypotheses[4] that are related to seed centred design. The result is summarised below.

- *Collective mind*

  1. *Chaos and complexity in structural design environments do not contribute to waste reduction.*
     Not falsified (not in prototype scope). Chaos produces dense orbits, which implies that SDP with many small mutations will appear. Complexity ensures that SDP offspring will be cross-pollinated within new structural design projects. This was not tested explicitly within the prototype but arguably, a portion of structural designer's dense orbits and cross-pollinations will appear in CloudConstruct and become available to fellow structural designers - which is the starting point for waste reduction. In order to falsify this hypothesis, the prototype must be extended to

[4] The null hypotheses can be falsified if there is statistical evidence for the opposite. For example, the null hypothesis "all swans are white" may be rejected if there is statistical evidence for the existence of swans with other colours.

maintain transition sets or other measurements to measure the effects of chaos and complexity in detail.

2. *An extremely easy to use HMI does not yield to more SDP.*
   Falsified. 90% of all structures in CloudConstruct were uploaded with the 'easy' method (auto-upload or QR code) while the remaining 10% were uploaded in the 'hard' way (open, create project, create structure).

- *Seed centred design*

  1. *Structural designers do not share SDP.*
     Falsified. 22% (+/-18%) of the questionnaire respondents indicated to publish their structures to CloudConstruct. 12% (+/-4%) of all projects were made public.
  2. *Structural designers do not use other structural designers' SDP.*
     Falsified. A total of 23% structures were copied, while 14% (+/-4%) were copied by structural engineers who did not own the SDP.
  3. *Structural designers use copied SDP unchanged.*
     Falsified. A comparison of the file contents shows that 5% (+/-4%) of all structures are exact copies. Therefore, the majority of the 23% copied structures include mutations or genetic manipulations.
  4. *The majority of SDP is one-of-a-kind.*
     Not falsified. 77% of all structures in CloudConstruct did not generate offspring, and therefore remained one-of-a-kind.
  5. *Structural designers remain independent of their own SDP.*
     Not falsified. The evidence to falsify this hypothesis comes from an unexpected direction. During the prototype testing, the author became engaged in the FrameDesign ecosystem to an extent that suggests that he became an RNA designer. This was not anticipated, since it was intended to find structural designers who were 'RNA engaged' with their SDP. The prototype did not offer enough functionality (i.e. a market environment) to prove the latter. Therefore this hypothesis cannot be falsified, but there is an indication that it may be falsified by future research.

6. *The structural design environment itself is not subject to evolution.*
   Falsified. After FrameDesign was published, many people (800+) rated it. The author corresponded with ~120 different users about bugs, improvements, translations and such for the FrameDesign product family. Together with the existence of strong competitors on the Android Market, became a driving force for the author to evolve the prototype (e.g. upgrading the initial 1D beam design into a 2D frame design tool) in order to remain attractive to users.

7. *Libraries do not emerge from SDP.*
   Falsified. FrameDesign is pre-installed with a single steel section (HEA120). In order to use additional sections, users must import them from a secondary app (Engineering Libraries) or input a custom section numerically. In 17% (+/-4%) of all structures, at least one additional section was added; there were at least 55 different section types. This information, together with future structures, may be used to maintain an evolving profile library.

The results in this section must be nuanced:

- The null hypotheses, the sampling source, the permitted margin of error, the rejection criteria etc were not formulated before the prototype was designed, due to the iterative nature of this research. Therefore, the null hypotheses testing is formally invalid. However the results provide incentives for for future research.

- The null hypotheses were formulated with SDP and structural designers in mind. However the data only included structures, which is a small subset of all possible SDP. Moreover, not all users were structural designers (44% professionals, 44% students, 12% others - Section 3.2.6).

- A better prototype design that includes more aspects of the theory (i.e. support for chaos and complexity, a reward system, CBR and a virtual assistant) may reveal entirely different results. This prototype was focussed on evolutionary development of structures.

- Not all uploaded structures may be representative, since there are indications that several users published test data only.

4.5 Limitations

A SEAM environment has its specific drawbacks that must be taken into account. Examples are:

- Shared, unsupervised products are not always traceable or reliable. Therefore structural designers remain responsible for their own designs. A SEAM will only assist by providing quality indicators and design proposals.

- It was assumed that it will be extremely easy for structural designers to use SEAM within their complex, chaotic real world environment. While humanoids and robots are being developed to operate in such environments, none of them are yet suitable and cost-effective virtual assistants compared to humans.

- SEAM may only be of use in environments in which a fair amount of repetition occurs.

- The theory in this thesis was limited to structural design. If it is applied to other AEC professions such as those in manufacturing and maintenance, then the theory may need to be revisited.

4.6 Advantages

1. The main advantage is the human centric approach. Artificial Intelligence will play an important role, but it is continuously being enriched with structural designers' unconscious knowledge.

2. There is less need for structural designers to collect and document their tacit knowledge, since a SEAM uses the cases themselves to infer its knowledge.

3. Certified SDP (i.e. a library) with references to actual projects will evolve, which eliminates the need to impose and maintain libraries.

4. Structural designers who create successfully evolving SDP will be visible as experts within the SEAM.

5. With a sufficiently strong mind and extremely easy to use HMI, the SEAM may function as an autonomous virtual assistant in the future.

6. Learning material may be produced from the SEAM. It may be relatively easy to extract concepts, illustrated with many cases as teaching material for structural designers.

7. Newly employed structural designers may easily find widely used SDP in order to become acquainted with the enterprise's particular approach to structural design problems.

4.7 Effect on the professional world of AEC

The focus within this research was specifically on structural designers in the Netherlands. However, many other professions within the AEC industry may benefit from socio-evolving products as well (e.g. structural engineers, innovators, students).

The application of the theory in this thesis may lead to more repetition without the need for unification. Therefore, the current (costly) initiatives in which it is attempted to achieve uniformity may be complemented with new SEAM environments in which participants may generate revenue and contribute to standardisation simultaneously. If such SEAM environments are used en mass, then it could have an impact on how AEC enterprises arrive at new designs. For example, current competitive advantages between AEC enterprises may dissolve and force AEC enterprises to innovate[5].

A SEAM environment may stimulate designers to focus on producing well-designed products with which they can generate revenue. In turn, AEC enterprises may start buying such products or hiring those designers rather than employing specialists, which may cause a new equilibrium between own experts and external SEAM designers[6].

[5] Some products that are assumed to be one-of-a-kind may prove to be repetitive within other enterprises.

[6] An example in this direction is Design Charrettes. These are public workshops in which the community is engaged in producing innovative solutions to societal issues.

4.8 Future research

This research might inspire other researchers to complement conscious developments with redundant, unconscious, evolutionary, tacit and chaotic aspects as found in Complex Adaptive Systems. The author would like to suggest a number of areas for future research.

- Using the as-is theory in order to implement a more sophisticated SEAM (e.g. with a reward system and CBR). More structural designers may benefit from the new SEAM. Also the new data may be useful for verifying the results of the current research as well as for expanding the theory.

- Including more aspects of the metaphor with nature such as meiosis, conception, protein production, protein transport, embodiment, reuse, illness, nutrition, immune systems and such. An organism life-cycle (conception to recycling) may be comparable to an AEC product life-cycle (inception to reuse).

- Implementing conflict handling mechanisms in a SEAM.

- Investigating a SEAM in relation to various other virtual enterprises within the AEC industry. Examples are architectural design, alliances, tenders, actual construction and other SEAM.

- Expanding virtual assistants with more sophisticated HMI and CBR to allow for intelligent, independent, learning and social entities.

- Finding innovative ways for DSS (such as a SEAM) to capture design intent.

APPENDIX A

Elaborating on preliminary prototypes

> "As soon as anyone starts telling you to be 'realistic', cross that person off your invitation list" – John Eliot

THis section describes the prototypes that were developed at a preliminary stage. They proved difficult to implement or not to reach enough designers within a limited time. The relation between these prototypes is shown in Figure 3.1.

A.1 Information at the source (IATS)

The Information At The Source (IATS) prototype (Figure A.1) is a Leviathan that gathers useful information from structural designers, attempts to induce context (e.g. from a file location to find the active project), and provides relevant CBR as well as certified product information (similar to Google Now). It is intended to be attractive and opaque.

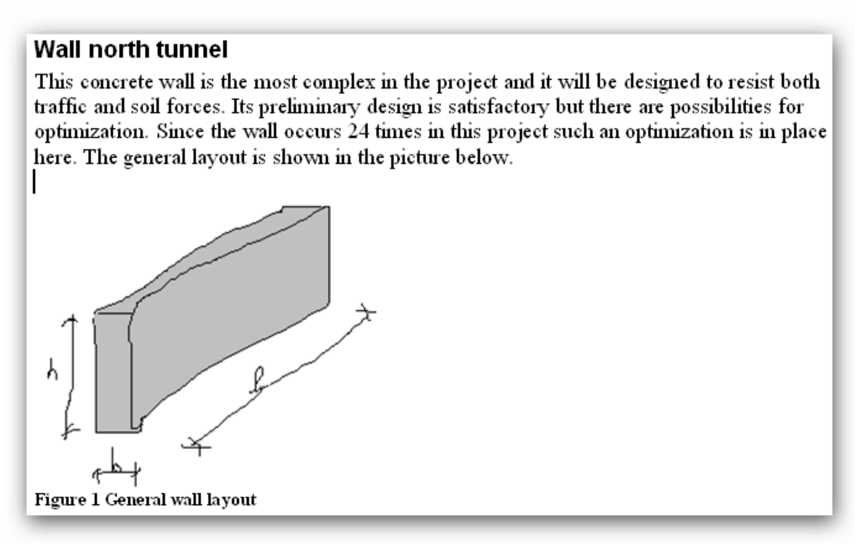


Figure A.1: *The IATS system integrates into the end users' tools and gathers and provides information with a minimum of user distraction. Here, a structural designer is reporting his calculations in a Word document, and IATS provided him with a picture and a description that it induced from the file's directory and the heading name.*

A.2 BIM Dashboard

A possible implementation for the conscious is to create a BIM dashboard. The dashboard implementation here is a web-based interface as illustrated in Figure A.2 with various gadgets for 3D visualisation, systems engineering, document management, cost & risk analysis and more, that interact with back-end systems. In order to couple the 3D model with SE, the draftsmen add the Systems Engineering (SE) code to the objects they draw. Since the SE breakdown structure uses the same SE codes, the loose coupling is a fact; licking on a node causes the objects to be highlighted and vise versa. A similar approach is valid for planning, documents and other project information.

A.3 Controlling a mouse cursor

The goal of this experiment is to demonstrate loose coupling in mouse control - that is, without an explicit connection between the prototype and the

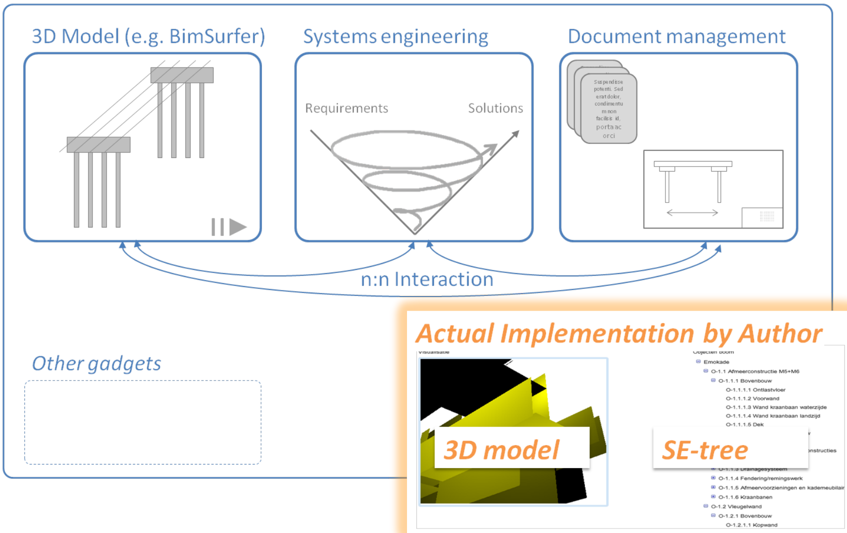


Figure A.2: *The BIM dashboard represents the conscious in a project while back-end systems and experts continue to provide the powerful unconscious. The inset shows the web-based prototype implementation. Clicking a node in the SE tree shows the corresponding objects in the 3D model and vice-versa.*

computer. The inspiration for a solution came from a brief observation of how human subjects find a mouse cursor if they do not know its position. They randomly move the mouse about, catch the movement and then direct the cursor to some goal, while gradually increasing precision in their motion in line with Fitts' law (Fitts (1992)). Obviously the computer screen must be identified and non-relevant movements such as running movies and people walking in the room must be neglected.

The implemented system looks for movements in the visual field (using background subtraction between two frames) while slightly moving the mouse in random directions. If it detects a movement, it checks whether that movement listens to deliberately chosen mouse movements. If this is true then the mouse cursor is considered to be found. For demonstration purposes the general "goal" was to move the cursor more or less diagonally up the screen. If the system loses track of the cursor while moving it towards the goal, the identification procedure restarts. A typical trajectory is shown in Figure A.3.

Figure A.3: *A typical trajectory (enlarged) produced by the mouse cursor program. Initially the system moves the mouse randomly until the system detects a movement. It then moves the cursor into deliberately chosen directions; if the detected movement listens to those directions, then the mouse is considered ´found´. If it loses the cursor then the procedure restarts.*

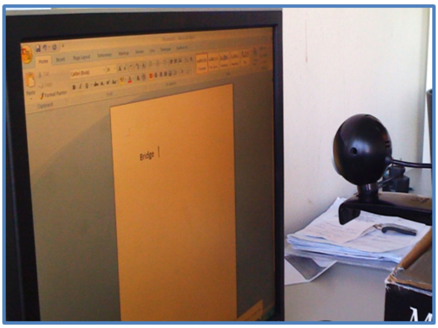


Figure A.4: *Prototype by master student Faridaddin Vahdatikhaki in which he attempted to make the corporate unconscious available to users in a straightforward way. [used with permission]*

A.4 Virtual assistant

As an offshoot of this thesis, under supervision of the author, the master student Vahdatikhaki (2011) elaborated on making knowledge bases available to users through natural HMI. He created a prototype with RoboRealm in which a webcam watches a computer screen, recognises captions such as "bridge" in a Word document and inserts appropriate pictures. There is a number of important concepts in this prototype. First, the solution enters the human perceptual world by using a vision system in the real world with all of its messiness and complications. Secondly, for inserting pictures it uses the tools that a human user would use: the computer screen, a mouse and a keyboard (Figure A.4). These two concepts allow the solution to operate any computer running any operating system, since there are no explicit (tight) couplings between the prototype and the computer.

APPENDIX B

Epigenetics

"Because things are the way they are, things will not stay the way they are" - Bertolt Brecht

Since Darwin time it is believed that the most fit-for-purpose organisms survive in a changing world, based on mutations or cross-overs that *happen* to be beneficial (Gould (2002)). Even though evolution is not goal-oriented[1], it is known that micro-evolution happens faster than expected in response to a changing environment. Examples are

- Bell et al. (2004) who describe how the phenotype of fish in the Loberg Lake, Alaska changed within 12 years instead of the expected millions of years
- the well-known pepper and salt moths that changed colour in response to air quality (and consequently their visibility for predators).

Unlike Darwin, Lamarck believed that an organism could pass certain acquired properties to its offspring (adaptive mutations), but his view was soon deprecated. However scientists are now discovering mechanisms

[1] Natural selection is a passive, random occurrence.

that do actively affect DNA. This controversial field of Epigenetic Programming is successfully being explored by for example Stockwell et al. (2003). Monk (1995) says that "Adaptive epigenetic / genetic inheritance challenges the 'central dogma' that information is unidirectional from the DNA to protein and the idea that Darwinian random mutation and selection are the sole mechanisms of evolution".

In the AEC industry, there is a strong form of epigenetics. Designers who actively improve products go beyond natural selection. Severe changes made by a designer in the product's embryonic DNA may be considered as either epigenetics, genetic manipulation or gene therapy[2] in more mature products[3].

---

[2] Designers sometimes call projects with too tight planning "Design during construction"

[3] Extreme conditions such as strong radiation also change an adult's genome (not as a whole but in random genes) and as a result the offspring may severely deviate from the ancestor.

APPENDIX C

Epitomes: Details hold assemblies

"Man is a physical and spiritual epitome of the Universe" - Daniel D. Palmer

IN biological systems, parts and details contain information about the whole. The more one goes into details, the more generic principles are associated with them (there are no endless decompositions of the same type). The author found this to be a very useful concept to build adapting definitions upon, and produced examples such as arbitrary mathematical expressions and structures that exhibit more than simple self-recursion (van de Ruitenbeek (2003)). This section provides an overview of epitomes and the relevance for the AEC industry.

C.1 Tree structures in nature

Assume that a real tree's root node is where the stem meets the ground. From there the stem grows and decomposes into branches, leaves, stems, petioles and leaflets, a fine vein structure, then individual cells and their components.

At that point nature has a surprise. One may expect increasingly simple building blocks, but then DNA appears. This compressed source of information holds the entire tree's blueprint, such as the tree's properties, how the tree should unfold from seed to tree, how it should reproduce and more. Some of this information is explicit, some of it implicitly depends on the environment that has been caused by itself (refer to Caporale (2006)).

Such rich sources of contextual information that are kept within details is what the author refers to as "epitomes". At an even a lower level, the atoms in a cell exaggerate an epitome since they expose similarities with galaxies (Figure Figure C.1 on page 73, also refer to Gentner (1983), who demonstrated how useful it is to make analogies like solar/atom systems). The idea of a galaxy within an atom is eloquently illustrated in the website provided by Matthew J. Parry-Hill and Davidson (2009) based on work by Boeke and Compton (1957). The further we understand decomposition/epitomes (e.g. the Higg's particle) the more overwhelming the entire system might prove or the more we understand it. For example, various formulas for concrete calculations (e.g. to determine crack width) are derived from insights in the material's molecular structure.

Ferns are textbook examples of normal endless recursion, but in fact they reveal epitomes. Consider the fern as a whole and one of its leaves. The leaf is a lookalike of the entire fern. The leaf consists of multiple smaller leaflets that are, again, fern lookalikes but this time crowded in a "forest". Going into more detail does *not* endlessly reveal more and more finer leaflets, but DNA, that holds information about the entire fern, its history, its possibilities to unfold in various conditions and so forth.

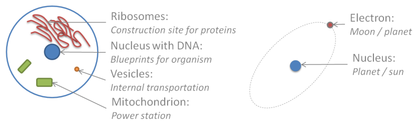
The principle is simple: lower levels of details refer to increasing higher levels. In ancient words: "Le bon Dieu est dans le detail"[1] (Gustave Flaubert, 1821-1880).

It is remarkable how low levels seem to correspond with high level systems. Examples are:

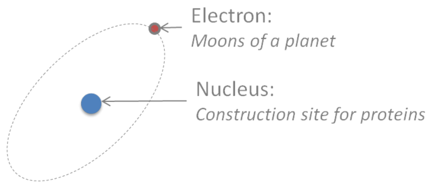
1. Cellular processes (waste disposal, reproduction, etc) match similar processes in a wood or a city;

2. DNA compresses individuals' and a nation's blueprint, including their ability to build a city and to create blueprints for designs;

[1] A popular variant is "The devil is in the details"



(a) *A seemingly low-level cell exhibits many high level processes: Construction, transportation, information management, waste disposal, power supply and more. Such processes can also be found on a construction site or in a city. [Source: author]*



(b) *Drilling into the atoms of a cell reveals that the atom model is a galaxy replica. Unraveling electrons or nuclei might therefore provide us with valuable information about the universe. [source: author]*

Figure C.1: *Beyond fractals: nature seems to apply more than simple recursion; each new detail level exposes references of a greater magnitude than the precursor level. [source: author]*

3. Atoms are analogous to solar systems and provide the means to build a DNA;

The Greek view was that the midpoint is Man, who summarises the cosmos.

A tree view does not only diverge; eventually it converges. Leaves touch one another and branches strangle to form a coherent whole. They might converge with one or more other trees or objects. The same is valid for swarms and flocks of animals. Therefore it is important to note that one cannot simply keep decomposing systems, since new levels represent new bodies of computation and therefore interact differently with their environment.

A final observation is that the root node is not the topmost node, rather it is the nucleus of a coherent set of incoming roots and outgoing branches, and in this sense it has similarities with nerve cells (dendrites, nucleus and axons).

C.2 Different types of recursion

A regular treeview decomposition algorithm (e.g. to produce a directory hierarchy) typically implements linear recursion routines that search or grow the tree. The UML notation is straightforward: any node has 0..* child nodes; typically those 0..* child nodes are *new*, lower-level nodes.

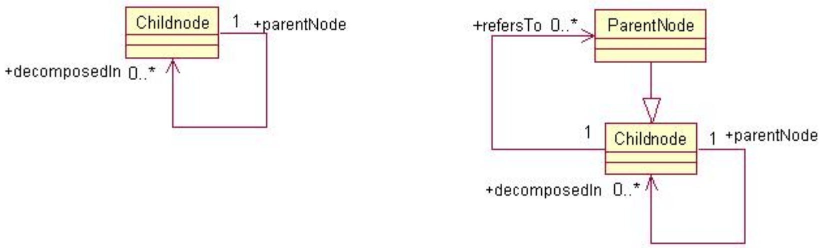


Figure C.2: *Both models presented here can represent epitomes, except that the second model stresses the concept. An epitome allows a child node to be either an existing (higher order) node or a new, lower level node. [source: author]*

Formally the UML model for an epitome is equal to that of linear recursion (refer to figure C.2). The subtle difference is that child nodes in epitomes are not only *new* lower level nodes, but also *existing* nodes of equal or higher order. A child node's 0..* children can therefore decompose into any other existing or new node within the system, including their own precursor - therefore the child node implicitly includes itself similar to the way a leaf in a tree refers to the entire tree through DNA.

The reference is not necessarily to existing nodes (instances) but the nodes may also include class definitions, that is, nodes that are potentially instantiated, depending on the particular. Most notably epitomes allow to compose both increasing complexity and increasing simplicity.

Within mathematics there is a variety of recursion forms:

- linear recursion - a function calling itself
- tail recursion - a function calling itself at the end (can easily be rewritten into a loop)
- binary & exponential recursion - a function calling itself twice & more
- nested recursion - a function who's argument is the function itself (e.g. Ackermann function)
- mutual recursion - function calling either themselves or each other
- primitive recursion - defines a new function in terms of existing, simpler ones. It uses a zero function, a successor function and projection functions, composition and linear recursion.

An epitome defines new functions in *more complex* terms than itself, and the recursion often occurs in an argument. Therefore an epitome can be classified as the amalgam of nested recursion, mutual recursion and the opposite of primitive recursion.

Where classical recursion forms yield a normal tree view, an epitome is a tree that roots and exposes, and increases complexity towards the end nodes which, as will surface in Appendix C.4.2, extends beyond mere data.

C.3 General mathematical formula using epitomes

Let $\langle Expression \rangle$ be any mathematical expression such as $1+\frac{1}{x}$. A $\langle Fraction \rangle$ can be composed of two $\langle Expressions \rangle$:

$$\langle Fraction \rangle = \frac{\langle Expression \rangle}{\langle Expression \rangle} \tag{C.1}$$

Since an $\langle Expression \rangle$ can contain other complex $\langle Fractions \rangle$, $\langle Fraction \rangle$ is a subset of $\langle Expression \rangle$ (refer to C.1). Conversely, the arguments within a $\langle Fraction \rangle$ can be any $\langle Expression \rangle$ including $\langle Fraction \rangle$ itself.

This is the essence of an epitome. It refers to its own precursor, which includes itself. It therefore permits complex nested fractions with increasing complex expressions in the arguments, such as

$$\frac{\left(\frac{\langle Expression \rangle}{\langle Expression \rangle}\right)}{\left(\frac{\langle Expression \rangle}{\langle Expression \rangle}\right)} \tag{C.2}$$

$\langle Expression \rangle$ can be extended with other child nodes such as π, $\sqrt{\langle Expression \rangle}$, $sin(\langle Expression \rangle)$, $\langle numbers \rangle$, $\langle variables \rangle$ etc. This simple definition allows for arbitrary mathematical expressions such as

$$\frac{\left(\frac{1}{cos(2/sin(3))}\right)}{\left(\frac{x}{3+tan(1/x)}\right)} \tag{C.3}$$

Apart from $\langle Expression \rangle$, the definition facilitates a full $\langle Equation \rangle$ such as $1+\frac{2}{4}=1,5$. Equation does not belong to $\langle Expression \rangle$ as to avoid invalid expressions such as $\frac{1+1=2}{2+2=4}$. The definition of $\langle Equation \rangle$ is:

Algorithm C.1 The definition of a common Formula using an epitome

$\underline{\langle Equation \rangle}$

Equation $= \langle Expression \rangle \langle Equalty \rangle \langle Expression \rangle$

$\underline{\langle Equalty \rangle}$

Equals $= =$

Does not equal $= \neq$

Greater than $= >$

Smaller than $= <$

$\underline{\langle Expression \rangle}$

$\langle Operation \rangle = \langle Expression \rangle \, \langle Operator \rangle \, \langle Expression \rangle$

$\langle Fraction \rangle = \frac{\langle Expression \rangle}{\langle Expression \rangle}$

$\langle Square \rangle = \sqrt{\langle Expression \rangle}$

$\langle Integral \rangle = {}^{\langle Expression \rangle}_{\langle Expression \rangle} \int \langle Expression \rangle$

$\langle Expression \rangle = {}^{\langle Expression \rangle}_{\langle Expression \rangle} \sum (\langle Expression \rangle)$

$\underline{\langle Trigonometry \rangle}$

sin$= sin(\langle Expression \rangle)$

cos$= cos(\langle Expression \rangle)$

tan$= tan(\langle Expression \rangle)$

$\underline{\langle Operator \rangle}$

Plus $= +$

Minus $= -$

Times $= *$

Divide $= /$

$\underline{\langle Number \rangle}$

1 = (any representation for '1')

2 = etc.

$$\langle Expression \rangle \, \langle Operator \rangle \, \langle Expression \rangle \tag{C.4}$$

The full formula definition is a simple tree view such as shown in C.1. A final result could be any full equation such as

$$\frac{\sqrt{sin^2(x)}}{3} \neq cos(\frac{1}{x}) \tag{C.5}$$

Epitomes are easily understood in terms of a $\langle Formula \rangle$ and functions that take an argument, such as $\sqrt{\langle Expression \rangle}$. But how can $\langle Expression \rangle$ itself refer to a precursor or maybe even higher level precursors? More in general, how can we use epitomes to represent anything arbitrary in the AEC industry, and what is its meaning?

To answer this question, consider a structural model $\langle Structure \rangle$ in

Algorithm C.2 The definition for arbitrary structures using epitomes. This definition allows one to use library elements but also existing structures to build up new structures. In essence every new instance becomes part of the library. This simple epitome definition allows for very complex structures to be grown.

$\underline{\langle StructuralElements\rangle}$

$\langle Beam\rangle = \langle Line\rangle$
$\langle Floor\rangle = \langle Plane\rangle$
$\langle Arbitrary\rangle = \langle Structure\rangle$

$\underline{\langle Structure\rangle}$

$\langle BeamOn2Supports\rangle = \langle Support\rangle\,\langle Structure\rangle\,\langle Support\rangle$

$\underline{\langle Support\rangle}$

$\langle Fixed\rangle = \langle Line\rangle\,\langle Line\rangle\,\langle Line\rangle\,\langle Line\rangle$
$\langle Hinge\rangle = \langle Line\rangle\,\langle Line\rangle\,\langle Line\rangle$
$\langle Arbitrary\rangle = \langle Structure\rangle$

$\underline{\langle Geometry\rangle}$

$\langle Point\rangle = \langle Number\rangle\,,\langle Number\rangle$
$\langle Line\rangle = \langle Point\rangle\,,\langle Point\rangle$

$\underline{\langle Number\rangle}$

1 = (any representation for '1')
2 = etc.

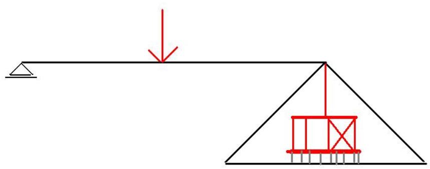


Figure C.3: *A simple beam can have two supports. The left support is a simple modelled support, the right support holds an epitome to ⟨Structure⟩ since any structure can function as a support. [source: author]*

terms of ⟨*Elements*⟩ (such as lines and planes), supports, loads and load combinations. Let an ⟨*Element*⟩ be attached to the precursor ⟨*Structure*⟩. ⟨*Element*⟩ holds lines, planes and such, which allows one to create any initial ⟨*Structure*⟩. A ⟨*Structure*⟩ contains a ⟨*Support*⟩. Since a ⟨*Support*⟩ is an epitome, this definition allows one to support to contain any⟨*Structure*⟩. A possible epitome representation is given in C.2.

⟨*Support*⟩ is an epitome, which allows a ⟨*Support*⟩ to be replaced by any ⟨*Structure*⟩. In practice this is very useful since one structure can support another (refer to Figure C.3 on page 77), but it may be modelled as a spring.

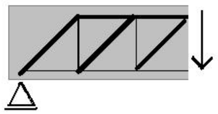


Figure C.4: *A larger structure-in-a-structure: typical epitome example of how smaller decomposition levels are richer than the precursor element.. The figure shows a concrete beam. The thick lines are compression lines, the thin lines are tension lines (reinforcement needed). In other words, the beam, which is a small element contains an entire truss structure. [source: author]*

A variation is a concrete beam. In order to calculate the amount of reinforcement it is often presented and calculated as shown in Figure C.4 on page 78. In that approach, there is an imaginary truss structure within the beam. The reinforcement bars take the tension forces while the compression forces go to imaginary concrete stripes. In other words, there is a complete structure (higher order) inside of the beam.

The definitions for$\langle Structure \rangle$ and $\langle Equation \rangle$ can be merged at a high level precursor. The highest precursor in the most general case is $\langle Anything \rangle$. In case of mathematical formulas, replace every occurrence of $\langle Expression \rangle$ with $\langle Anything \rangle$. In case of structures, replace every occurrence of $\langle Structure \rangle$ with $\langle Anything \rangle$. Consequently, the following expression is valid:

$$\frac{\langle Anything \rangle}{\langle Anything \rangle} \tag{C.6}$$

which allows one to divide a beam in two:

$$\frac{\langle Structure \rangle}{\langle Number \rangle} \; or \; \frac{Beam}{2} \tag{C.7}$$

The logic for actually dividing the beam in two is now outside of mathematics and structures. It is up to that epitome level (most probably a structural engineer) to decide on the division logic.

Another epitome example is a space. Designers tend to divide a building into spaces and subspaces. But consider to allow a subspace to not only hold subspaces but also precursor classes such as the building it belongs to. That leverages possibilities for the subspace; a subspace may then contain anything a building contains: subspaces but also a kitchen or a complete building-in-a-building or a city-in-a-room (e.g. in the form of a maquette or building plans).

Thinking of a building interior as being of a higher magnitude makes sense: a building interior is the miniature equivalent of the city it belongs to with its streets (corridors), suburbs (rooms), subways (basement), places to eat, sleep, work and relax, illumination at night, garbage collection and so forth. A similar comparison between a) the independently working "agencies" and their connections in the human brain versus b) cities, towns and highways was made by (Minsky, 1986, p. 314).

An Internet screen on a computer is also an epitome: the computer is supposed to be a low decomposition level in a house but one can contact the entire world from behind it. The same is valid for a radio or TV, expressions in music, memories, books etc. Also consider how scientists find a great deal of a star's properties by observing simple light particles it emits or how handwriting reveals part of someone's personality. Finally, parameters in equations of for example the new Eurocode are small elements of the theory but they represent a world of theories and experiments.

The human body is a remarkable example of epitomes. All body parts could be classified in a tree view down to the level of cells. That leads to the discovery that instead of finding elementary building blocks at the utmost leaf nodes, the cells contain the definition of the entire body, which enables every single cell to see itself in context of the entire body.

C.4 Epitome description with Complex Numbers

Epitomes are imaginary. They are included in real things and influence real things, but remain invisible themselves. They are definitions, not real things, even though real things such as acids in DNA might enable them to exist. Such a connection between a real thing and its imaginary part is the domain of Complex Numbers. This section describes how complex numbers may be used to define epitomes.

C.4.1 Complex numbers

In general, complex numbers may describe how input is related to response. An example is a mass-spring system that responds to an oscillating fixture (refer to picomonster (2012)). If the complex number that describes the output lacks an Imaginary part, then there is a purely Real response; that is, the response's high and low points are in sync with the oscillating fixture point. If an Imaginary part is added, then the response lags the output (e.g. due to a damper). Any combination of both describes a mass-spring-damper system. Lesurf (2010) similarly described electronic signals with complex numbers.

In an epitome tree, each node S is comparable with a complex number. The Real part represents actual values such as dimensions and loads. The Imaginary part is the epitome. For example, if S is a concrete beam, then R may be the beam dimensions and and a distributed load, while I is the imaginary internal truss structure (refer to figure C.4). The beam response (e.g. the deformation) lags the input values, since the internal truss structure distributes the loads internally.

Complex numbers are written in the form of $z = a + bi$. However in the context of epitomes, it is more convenient to write in the form of sets, since an epitome tree contains sets (nodes and subnodes). An epitome tree node S may be written as:

$$S = \{R, I\} \tag{C.8}$$

Since I is an epitome, S is a proper subset[2] of its own Imaginary part (e.g. a beam is a proper subset of a truss structure):

$$S \subsetneq I \tag{C.9}$$

Therefore, C.8 should be re-defined as

$$S \subsetneq \{R, I\} \tag{C.10}$$

The Imaginary part I refers to the powerset P (i.e. any combination of) of all precursors *P1,P2,P3...* These precursors are S and S's parent nodes (which includes S itself), as well as other equal level nodes.

$$I = P(\{P1, P2, P3...\}) \tag{C.11}$$

Substituting C.11 in C.10 finally gives:

[2] If B is a proper subset of A then none of B's elements are outside of A, and at least one of A's elements are outside of B.

$$S = \{R, P(\{P1, P2, P3...\})\} \quad \text{(C.12)}$$

This equation says that a node S in an epitome tree is composed of real values and a definition from any precursor. That is, any Real value can be complemented with an Imaginary, more complex model. In this regard, the reader may prefer to think of epitomes as a form of "ancestor recursion".

C.4.2 The smallest Real elements are epitomes

Fleshing out R and I leads to a contradictory conclusion. The real part R represents actual numbers, lines, points and operators. However those are non-existing (unreal!) human inventions, whereas the Imaginary part I refers to the set of real-world occurrences (e.g. complete but abstract structures) that humans can understand without having exact R representations.

A solution is to reverse R and I in the previous discussion. However that violates the general complex number concept that R defines direct response while I defines time-lagged response. Consider this in the light of a Leviathan. To the AEC system, Real actual projects and participants are necessary to produce response, but they are just as unreal as numbers. What Really matters is the Imaginary, epitomic Leviathan. It stabilises (time-lags) the AEC industry with its body of construction knowledge.

Only the composite node S is meaningful when building epitome tree views. Omitting either R or I yields meaningless structures. Omitting R implies creating purely imaginary structures (such as fully parametrised structures, attached to relevant theories and previous knowledge) while omitting I implies creating one-of-a-kind products without any theoretical backbone. Arguably, a full complex structure includes both Real and Imaginary values.

The richness of an S composition is the anti-pole of a regular tree view decomposition that lacks this description power. A regular tree view decomposition is R explicit at every level. Think of a computer application that calculates a foundation in great detail, without being able to interpret the R values it generates in a report afterwards. S prevents such ignorance. For instance, without specific recalculations, S is aware of possible project cost consequences (I) if a certain pile sheet calculation uses a too optimistic δ value (R).

Nomenclature

Abduction Assuming that something caused an observed effect or assuming that a certain solution will solve a problem (but it remains an hypothesis). This is the opposite of induction, in which one may prove based on evidence that a certain cause has a certain effect in most cases.

AEC industry The Architectural, Engineering and Construction industry

BIM Building Information Modelling is the vision that all involved actors in a project can flawlessly cooperate on a centralized, properly managed building model - regardless of their language, role or location.

CAS A Complex Adaptable system is often a living system (such as an ant colony) that continuously changes its behaviour and its social network, based on previous experience. The AEC industry is an example of a CAS.

Cloud An online, on-demand storage or service.

Cognitive entity A (virtual) being, capable of acquiring knowledge and understanding through thought, experience, and the senses [Oxford Dictionary]

Conscious The conscious is that part in the mind that contributes to self-awareness and rationality.

Design swamp The diffuse set of tools in the AEC industry that continuously appear and disappear. They are neither high-level or low-level tools, and therefore not general-purpose.

Designerly A term coined by design researcher N. Cross. It may be translated as The intrinsic nature of a designer.

DNA DeoxyriboNucleic Acid is the carrier of genetic information of living organisms. In this research, DNA is used as the equivalent of all information (both implicit and explicit) that is needed to produce the product. RNA (designers) translate this information into actual building blocks.

Enterprise Factual and / or virtual entities in any form of collaboration for an unspecified amount of time. An example is humans and virtual personalities that collaborate in a virtual and / or or factual design environment.

Epitome A thing representing something else in miniature

Genetic manipulation An imposed change in a design. Designers often apply genetic manipulation, since natural selection is passive and undirected.

Hybrid agents One may refer to hybrid agents if a design environment constitutes of both artificial and human intelligence that provide relevant knowledge.

Leviathan A superior creature with whom one could exchange his own freedom for protection. Something big that exists for the benefit of the masses.

Mind Combination of the conscious and the unconscious

Organism An instantiated form of information evolution and mind.

RNA Ribonucleic Acid translates information in DNA into actual proteins. In this research, RNA is the equivalent of designers who translate general product information into actual, specific designs.

Unconscious The unconscious manages the thoughts that are out of reach of the conscious; it is impossible to choose to bring them to the conscious (therefore strictly speaking, the author cannot provide an example here). The subconscious however can be reached by the conscious (e.g. a phone number).

Virtual enterprises Conglomerates of AEC actors that appear and dissolve on a project basis.

Wicked problem Problems that have complex interdependencies and incomplete, changing and contradicting requirements.

Bibliography

B.J. Baars. In the theater of consciousness:: The workspace of the mind. 1997.

R. Beheshti. Design decisions and uncertainty. *Design Studies*, 14(1):85–95, 1993. ISSN 0142-694X.

M.A. Bell, W.E. Aguirre, and N.J. Buck. Twelve years of contemporary armor evolution in a threespine stickleback population. *Evolution*, 58(4): 814–824, 2004.

B.C. Benne. *Managing AEC project organizations at the edge of chaos: An analysis of AEC projects' adaptive capacity from a living systems perspective.* PhD thesis, University of California, Berkeley, 2006.

M.R. Bennett. *Neuroscience and philosophy: Brain, mind, and language.* Columbia Univ Pr, 2007.

BLS. Employment in construction - bureau of labor statistics, 2010. URL `http://www.bls.gov/iag/tgs/iag23.htm`.

K. Boeke and A.H. Compton. *Cosmic view: the universe in 40 jumps.* J. Day, 1957.

Bouwkennis. Bouwkennis, 2011. URL `http://www.bouwkennis.nl/producten/bouwkennis_jaarrapport_nederland,http:`

//www.bouwkennis.nl/pers/bouw_zoekt_oplossing_faalkosten_in_uitvoerbaarheid_ontwerp/faalkosten.

D. Burford and E. Blake. Face-to-face implies no interface. *2nd South African Conference on Human-Computer Interaction (CHI-SA2001)*, 2001. URL http://people.cs.uct.ac.za/~dburford/masters/techreports/CS01-15-00/CS01-15-00.pdf.

L.H. Caporale. *The implicit genome*. Oxford University Press, USA, 2006.

R. Clarke. Asimov's laws of robotics: Implications for information technology. *Computer*, 27(1):57–66, 1994.

N. Cross. Designerly ways of knowing. *Design studies*, 3(4):221–227, 1982.

CROW. Cheobs object libraries, 2008. URL http://www.crow.nl/cheobs.

CROW. Visi, 2012. URL http://www.crow.nl/nl/Online_Kennis_en_tools/VISI/VISI-VISI.html.

CUR. Building objects and virtual construction, 2008. URL http://www.coinsweb.nl/.

Edwin Dado. *ICT-Enabled Communication and Co-Operation in Large-Scale On-Site Construction Projects*. PhD thesis, Technische Universiteit Delft, 2002.

K. Dautenhahn. Getting to know each other - artificial social intelligence for autonomous robots. *Robotics and autonomous systems*, 16(2):333–356, 1995.

D. Davis. Lean, green and seen. *Journal Of Building Information Modeling, Fall*, 2007.

S. DeDeo, D.C. Krakauer, and J.C. Flack. Inductive game theory and the dynamics of animal conflict. *PLoS computational biology*, 6(5):e1000782, 2010.

AJ Dijksterhuis. Het slimme onbewuste: denken met gevoel. 2008.

K.J. Dooley. A complex adaptive systems model of organization change. *Nonlinear Dynamics, Psychology, and Life Sciences*, 1(1):69–97, 1997.

B.T. Dotta, C.A. Buckner, R.M. Lafrenie, and M.A. Persinger. Photon emissions from human brain and cell culture exposed to distally rotating magnetic fields shared by separate light-stimulated brains and cells. *Brain research*, 2011.

N. Eldredge and S.J. Gould. Punctuated equilibria: an alternative to phyletic gradualism. *Models in paleobiology*, 82:115, 1972.

Eurocode. Eurocode, 2012. URL `http://www.eurocodes.nl/`.

L. Fahey and L. Prusak. The eleven sins of knowledge management. *California management review*, 40(3), 1998.

G. Fischer, A.C. Lemke, R. McCall, and A.I. Morch. Making argumentation serve design. *Human-Computer Interaction*, 6(3):393–419, 1991.

G. Fischer, R. McCall, J. Ostwald, B. Reeves, and F. Shipman. Seeding, evolutionary growth and reseeding: supporting the incremental development of design environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, pages 292–298. ACM, 1994. ISBN 0897916506.

P.M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology: General*, 121(3):262–269, 1992.

M.J. French. *Invention and evolution: design in nature and engineering*. Cambridge Univ Pr, 1994.

D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170, 1983.

J. Goldstein. Emergence as a construct: History and issues. *Emergence*, 1 (1):49–72, 1999.

S.J. Gould. *The structure of evolutionary theory*. Belknap Press, 2002.

H. Gruber and P. Koutroumpis. Mobile communications: Diffusion facts and prospects. *Communications and Strategies*, 77(1):133–45, 2010.

B. Hasselblatt and A.B. Katok. *A first course in dynamics: with a panorama of recent developments*. Cambridge Univ Pr, 2003.

T. Hobbes, Inc Infomotions, Virginia Polytechnic Institute, and State University. *Leviathan, 1651.* Scolar P., 1969.

John H. Holland. *Emergence: From Chaos to Order.* Perseus Publishing, 1999. ISBN 0738201421.

L. Horváth and I.J. Rudas. Human intent representation in knowledge intensive product model. *Journal of Computers*, 4(10):954–961, 2009.

S. Hübner. Building a learning organization. *Harvard Business Review*, 75 (5):148, 1995.

I. Hybs and J.S. Gero. An evolutionary process model of design. *Design Studies*, 13(3):273–290, 1992.

A. Juarrero. Dynamics in action: Intentional behavior as a complex system. *Emergence*, 2(2):24–57, 2000.

C.G. Jung. The concept of the collective unconscious. *Collected works*, 9 (Part I), 1936.

JM Kamara, G. Augenbroe, CJ Anumba, and PM Carrillo. Knowledge management in the architecture, engineering and construction industry. *Construction Innovation: Information, Process, Management*, 2(1):53–67, 2002.

H. Kim, G. York, G. Burton, E. Murphy-Chutorian, and J. Triesch. Design of an anthropomorphic robot head for studying autonomous development and learning. 4, 2004.

EI Knudsen and MS Brainard. Creating a unified representation of visual and auditory space in the brain. *Annual Review of Neuroscience*, 18 (1):19–43, 1995. URL `http://jacknife.med.yale.edu/nsci590/pdfs/knudsen1995.pdf`.

J. Kolko. Abductive thinking and sensemaking: The drivers of design synthesis. *Design Issues*, 26(1):15–28, 2010.

S. Lang. Cognitive factors in distributed design. *Computers in Industry*, 48(1):89–98, 2002. URL `http://people.arch.usyd.edu.au/~adong/courses/desc9097/assets/lang-cognitivefactorsdistributeddesign.pdf`.

R. Leifer. Radical innovation: How mature companies can outsmart upstarts. 2000.

Jim Lesurf. More complex makes simple, 2010. URL `www.st-andrews.ac.uk`.

S.A. Levin. Ecosystems and the biosphere as complex adaptive systems. *Ecosystems*, 1(5):431–436, 1998.

B. Libet, C.A. Gleason, E.W. Wright, and DK Pearl. Time of conscious intention to act in relation to onset of cerebral activity (readiness-potential). *Neurophysiology of consciousness: selected papers and new essays*, page 249, 1993.

D. Loorbach and R. van Raak. Governance in complexity: An multi-level policy-framework based on complex systems thinking. In *Paper for the conference'Lof der Verwarring', Congresbundel*, volume 19, 2005.

Christopher A. Burdett Matthew J. Parry-Hill and Michael W. Davidson. Secret worlds: The universe within, 2009. URL `http://micro.magnet.fsu.edu/primer/java/scienceopticsu/powersof10/`.

M. Minsky. *The society of mind.* Simon & Schuster, Inc. New York, NY, USA, 1986.

MIT. Open source building alliance operation (osba), 2011. URL `http://p2pfoundation.net/Open_Source_Building_Alliance_Operation`.

M. Monk. Epigenetic programming of differential gene expression in development and evolution. *Developmental genetics*, 17(3):188–197, 1995.

I. Nonaka and H. Takeuchi. *The knowledge-creating company: How Japanese companies create the dynamics of innovation.* Oxford University Press, USA, 1995.

G. Paperin, D.G. Green, and S. Sadedin. Dual-phase evolution in complex adaptive systems. *Journal of The Royal Society Interface*, 8(58):609–629, 2011.

picomonster. Applications of imaginary numbers in the real world, 2012. URL `http://www.picomonster.com/`.

Cyrus Pinto. How autonomous vehicle policy in california and nevada addresses technological and non-technological liabilities. *Intersect: The Stanford Journal of Science, Technology and Society*, 5(0), 2012. URL `http://ojs.stanford.edu/ojs/index.php/intersect/article/view/361`.

SN Pollalis. Beyond islands of automation. *BOSS Magazine*, 4:28–29, 1997.

Yacine Rezgui, Christina J. Hopfe, and Chalee Vorakulpipat. Generations of knowledge management in the architecture, engineering and construction industry: An evolutionary perspective. *Advanced Engineering Informatics*, 24(2):219 – 228, 2010. ISSN 1474-0346. doi: 10.1016/j.aei.2009.12.001. URL `http://www.sciencedirect.com/science/article/pii/S1474034609000810`. <ce:title>Enabling Technologies for Collaborative Design</ce:title>.

E.L. Rissland. Ai and similarity. *Intelligent Systems, IEEE*, 21(3):39–49, 2006.

H.W.J. Rittel and M.M. Webber. Dilemmas in a general theory of planning. *Policy sciences*, 4(2):155–169, 1973.

B.J. Ritzel, J.W. Eheart, and S. Ranjithan. Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. *Water Resources Research*, 30(5):1589–1603, 1994.

J.J. Rousseau. *The social contract.* Wiley Online Library, 1947.

L.L. Salvador. The mechanical and universal nature of imitation. *Agents in Interaction*, 1998.

R.C. Shank. Dynamic memory. *Cambridge University, New York*, 1982.

C.A. Stockwell, A.P. Hendry, and M.T. Kinnison. Contemporary evolution meets conservation biology. *Trends in Ecology and Evolution*, 18(2):94–101, 2003.

Sybase. Sybase, 2011. URL `http://www.Sybase.com`.

C. Taylor and D. Jefferson. Artificial life as a tool for biological inquiry. *Artificial Life*, 1(1_2):1–13, 1993.

E. Tiezzi. *Steps towards an evolutionary physics*, volume 15. Wit Pr/Computational Mechanics, 2006.

Faridaddin Vahdatikhaki. Loosely-coupled user-friendly interface: a window toward a better application of knowledge-based system in architecture, engineering and construction. Master's thesis, Delft University of Technology, 2011.

H.K.M. van de Ruitenbeek. Handschrift- en tekeningherkenning voor constructeurs. Master's thesis, Delft University of Technology, 2003.

S. Vosniadou and A. Ortony. *Similarity and analogical reasoning*. Cambridge Univ Pr, 1989.

W. Weaver. Science and complexity. *American scientist*, 36(4):536–544, 1948.

R.O. Weber, R.M. Barcia, A. Martins, and R. Pacheco. Using typicality theory to select the best match. 1996.

Curriculum vitae

Birth
26-08-1978, Johannesburg, South Africa

Secondary education
Atheneum, Veenlanden college Mijdrecht (1992-1997)

Tertiary education
Civil engineering, TU Delft (1997-2003).
Specialisation in Civil Technical Informatics. Master Thesis: Handwriting and Drawing recognition for Structural Designers.

PhD
TU Delft (2006 - 2012)

Professional experience
Specialist at Volker InfraDesign (2007-2012, current)
Principal designer at Spie Controlec engineering (2003-2007)
Software architect at Speechcom (2001-2002)

The AEC industry is a complex, distributed and chaotic system in which virtual enterprises conduct both repetitive and one-of-a-kind projects under the pressure of heavy competition. There is waste due to the sub-optimal organization, but the industry does continuously adapt to a changing world and keeps producing useful buildings and infrastructure. Therefore, the AEC industry has the characteristics of a Complex Adaptive System.

ISBN 978-90-808007-0-0

9 789080 800700 >