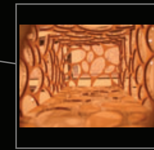
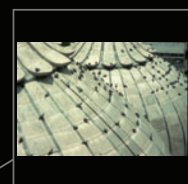
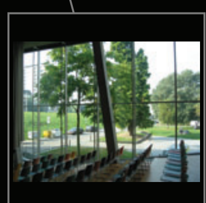
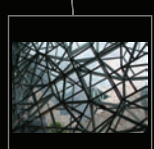
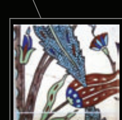








ISBN 978-90-8570423-2

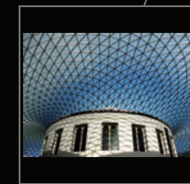
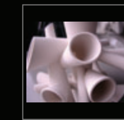
The Architectural Information Map

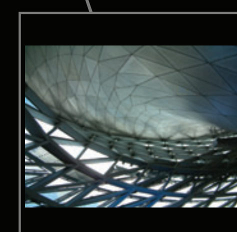
Semantic modeling in conceptual architectural design

Esma Bige Tunçer

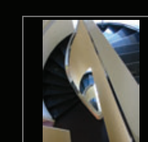
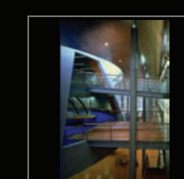
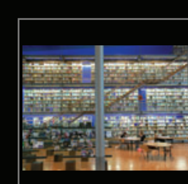
The Architectural Information Map

Semantic modeling in conceptual architectural design




Esma Bige Tunçer


The Architectural Information Map

Semantic modeling in conceptual architectural design

Esma Bige Tunçer

PROPOSITIONS

1. Precedent analysis in architectural education would radically improve through the application of ArcIMap.

2. Computers can design.

3. Associative networks suit perfectly well for structuring architectural information.

4. Going directly from A to B is efficient. Lingering on the way for gaining knowledge, experience and insights is priceless.

5. Designers aim at designing a masterpiece, but live in a characterless row house.

6. Subjectivity and individualism are essential to information classification in architecture. They stand in opposition to standardization.

7. Without curiosity there can be no innovation in science as well as architecture.

8. In academia one can't just "add women and stir" (Bunch, 1981).

9. Unintended uses of computational tools can be at least as valuable as intended uses.

10. For intended uses of computational tools, computational design support must be situated in its context in order to be successful.

11. Classification is a human trait. However, one should try to avoid classifying people in a discriminating manner.

12. The use of scientific methods and techniques in architecture does not guarantee a good building.

Bunch, C. (1981). Feminism in the 80's: Facing down the right. Denver, Inkling Press.

These propositions are considered opposable and defendable and as such have been approved by the supervisor Prof.dr.ir. I.S. Sariyildiz.

The Architectural Information Map

Semantic modeling in conceptual architectural design

Esma Bige Tunçer

STELLINGEN

1. De toepassing van ArcIMap zou de analyse van precedenten in het architectuur onderwijs radicaal verbeteren.

2. Computers kunnen ontwerpen.

3. Associatieve netwerken zijn uitermate geschikt voor het structureren van architectonische informatie.

4. Rechtstreeks van A naar B gaan is efficiënt. Dwalen onderweg voor het verkrijgen van kennis, ervaring en inzichten is onbetaalbaar.

5. Ontwerpers streven naar het ontwerpen van een meesterwerk, maar wonen in een karakterloos rijhuis.

6. Subjectiviteit en individualisme zijn van essentieel belang bij het classificeren van informatie in de architectuur. Deze staan in tegenstelling tot standaardisatie.

7. Zonder nieuwsgierigheid kan er geen innovatie in wetenschap én architectuur zijn.

8. In de academische wereld kan men niet alleen "vrouwen toevoegen en roeren" (Bunch, 1981).

9. Onbedoeld gebruik van computationele tools kan minstens even waardevol zijn als het beoogde gebruik.

10. Voor beoogd gebruik moet ondersteuning voor computationeel ontwerpen in zijn context gesitueerd worden om succesvol te zijn.

11. Classificeren is een menselijke eigenschap. Het classificeren van mensen op een discriminerende wijze dient men echter te proberen te voorkomen.

12. Het gebruik van wetenschappelijke methoden en technieken in de architectuur is geen garantie voor een goed gebouw.

Bunch, C. (1981). Feminism in the 80's: Facing down the right. Denver, Inkling Press.

Deze stellingen worden opponeerbaar en verdedigbaar geacht en zijn als zodanig goedgekeurd door de promotor Prof.dr.ir. I.S. Sariyildiz.

The Architectural Information Map

Semantic modeling in conceptual architectural design

The Architectural Information Map

Semantic modeling in conceptual architectural design

Proefschrift

ter verkrijging van de graad van doctor

aan de Technische Universiteit Delft,

op gezag van de Rector Magnificus prof.dr.ir. J.T. Fokkema,

voorzitter van het College voor Promoties,

in het openbaar te verdedigen

op 3 december 2009 om 10.00 uur

door

Esma Bige TUNÇER

geboren te Ankara, Turkije

Bachelor of Architecture, Middle East Technical University, Turkey

Master of Science in Architecture, Carnegie Mellon University, USA

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr.ir. I.S. Sariyildiz

Samenstelling promotiecommissie:

Rector Magnificus	Voorzitter
Prof. dr.ir. I.S. Sariyildiz	Technische Universiteit Delft, promotor
Prof. Dr. R. Krishnamurti	Carnegie Mellon University, USA
Prof. Dr. C.J. Anumba	Pennsylvania State University, USA
Prof. Dr.-Ing. G. Schmitt	Eidgenössische Technische Hochschule Zürich, Switzerland
Prof. dr.ir. B. de Vries	Technische Universiteit Eindhoven
Prof. dr.ir. J.C. Paul	Technische Universiteit Delft
Dr.ir. H. Zijlstra	Technische Universiteit Delft

Printed and Distributed by Wöhrmann Print Service

Delft, The Netherlands

ISBN 978-90-8570423-2

To my parents

Tuna and Süer

COVER IMAGE CREDITS


Roof of the British Museum Great Court, London, 2000, Foster and Partners and Buro Happold.
Photograph from http://www.panoramio.com/photo/6754278


Rapid prototype model of a lamp design by Jacques van Adrichem, Space and Perception Digital Manufacturing Workshop of the Delft School of Design, September 2007. Photograph by Bige Tunçer..

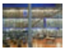

Kapel Heilige Maria der Engelen, Rotterdam, 2001, Mecanoo Architects.
Photograph from http://www.mecanoo.com/


BMW Welt, Munich, 2007, Coop Himmelb(l)au and Bollinger+Grohmann.
Photograph by Michela Turrin.


Interactive 3D interface for the ICCS environment, developed by Bige Tunçer, 1998, ETH Zurich.


Stairs inside the Sonneveld House, 1933, Rotterdam, Brinkman & Van der Vlugt.
Photograph by Dag Thielens.


Interior of the Central Library Delft University of Technology, Delft, 1998, Mecanoo Architects.
Photograph from http://www.mecanoo.com/


Interior of the Central Library Delft University of Technology, Delft, 1998, Mecanoo Architects.
Photograph by Bige Tunçer.


Kunsthal, 1992, Rotterdam, OMA.
Photograph from http://www.flickr.com/photos/gijsvanderwal/3702358384/, by Gijsbert van der Wal.

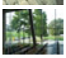

New Milan Fair, 2005, Massimiliano Fuksas and MERO-TSK.
Photograph courtesy of Dr. Jaime Sanchez Alvarez.


Laser-cut model of a mobile pavilion design by Astrid Nolte, Katharina Überschär andThomas Allen, International Design Studio SmartStructures, 2006.


Rapid prototype model selected from the elective course "Mediated Discourse as a Form of Architectonic Intervention," 2003, TU Delft.


Korykos: an ancient city in Cilicia, Anatolia, now Kızkalesi, Mersin, Turkey. The name Korykos was first mentioned in 197 BC. Photograph by Bige Tunçer.

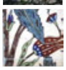

Detail from the domes of Jama Masjid (Friday Mosque), 1656, Delhi.
Photograph by Bige Tunçer.


Kunsthal, 1992, Rotterdam, OMA.
Photograph by Bige Tunçer.


Open hand sculpture, Chandigarh, 1954, Le Corbusier.
Photograph by Bige Tunçer.


A wall near the (late Hellenistic) Zeus Temple of the ancient city of Paperon, Anatolia, now Narlikuyu, Mersin, Turkey. Photograph by Bige Tunçer.


Federation Square, 2002, Melbourne, Lab Architecture Studio and Atelier One.
Photograph from http://www.flickr.com/photos/87791108@N00/2592036025/


An Iznik ceramic tile from the Selimiye Foundation Museum.
Photograph by Bige Tunçer.


Laser-cut model of a stadium design by Juan Manuel Dávila Delgado, Daan Font Freide, Rudolf van der Meulen, and Simon Bolle, XXL, TU Delft, 2007.

SUMMARY

This research focuses on the acquisition, representation, sharing and reuse of design information and knowledge in the conceptual phase of architectural design, and targets the creation of situated digital environments where teams of designers communicate and collaborate using this information and knowledge.

The main product of the conceptual architectural design phase is a design concept that promises to be successfully developed for the given design project. In order to foster this concept generation, designers gather information in order to gain knowledge and insights about a design task at hand, but also to get inspiration and creative ideas. Precedents – known examples of good design solutions – act as a common source of knowledge and inspiration for designers. Many precedent libraries exist where (visual) documents are collected in a repository, generally organized according to common categories such as 'year of completion' and 'architect'. However, in the conceptual design phase, designers are generally not ready to formulate specific queries for retrieving information. One may be interested in looking at all documents about a certain topic, or just jumping from link to link, following a certain thread. This requires that the information structure that relates the documents is dense enough and at the same time possesses an organizational structure that allows a categorization of documents that is more powerful than a simple collection of common categories. Furthermore, when designers define an organizational structure for the knowledge and insights they gain from precedents, this fortifies their design reasoning process. That is, designers construct a cognitive model of relevant connections between the current problem and the design rationale on the one hand, and the knowledge and concepts underlying precedents stored in the repository on the other hand. Therefore, the digital environments aimed at in this research enable their users to collectively, interactively and incrementally develop an information structure that organizes the information and knowledge residing in the environment.

A community of designers commonly shares a common professional language where the vocabulary of this language represents a shared understanding. This language is formed over time and passed on to new members of the community. Members of such a design community working together on a common goal (generally a project) form a community of practice. Members of a community of practice operate both by recording common knowledge into documents and by actively participating in social processes in order to personally contextualize this recorded knowledge. These activities are both the means and the result of an architectural community of practice. By performing both activities, members of an architectural community of practice collectively agree on the value of this

knowledge and information. This is denoted correspondence, i.e., communication with the aim of reaching an agreement. Correspondence ensures that the creators of the information and knowledge are also its users. Correspondence is the key to the creation of a dense and highly inter-related information structure that forms the basis of digital information environments for the conceptual phase of design. Such information structures are denoted 'complex information structures' in this research.

A complex information structure is composed of information entities and their relationships, tagged with certain design concepts. These design concepts are themselves related through semantic relationships forming a semantic structure. This semantic structure acts as the organizational backbone of a complex information structure. Elements (concepts and relationships) of this semantic structure are associated with information entities (documents) and describe them. A complex information structure created by a community of practice through information and social processes has characteristics of a 'complex adaptive system' where the structure of the system is non-hierarchical, the interactions are not predefined, and the state of the system is unpredictable.

In this research, grounded theory has been adopted as the research methodology in order to develop a context based and iterative approach to the research domain and issues. In this context, the research question has been iteratively formulated as: How can communities of architectural practice correspond on design information and knowledge during the conceptual phase of design? Case studies have provided empirical and qualitative data in order to ground and iteratively formulate the theory. This research question has been addressed through a study of relevant literature, theories, methods and techniques, and has led to the development of a computational framework called the Architectural Information Map (ArcIMap). Complex information structures form the basis of ArcIMap. The goal of ArcIMap is to define a structure for the design and creation of digital applications that support designers in the conceptual phase of design by defining the representational framework for achieving an integrated information structure of components, relationships and metadata from a collection of design documents and the knowledge that resides in these documents. The framework can then be implemented for different purposes, domains, contexts, or architectural bodies.

ArcIMap is both a method and a model. The method defines social and information processes in order to create complex information structures underlying complex adaptive systems. The model acts as a structure for the design of complex information structures. The techniques and technologies encapsulated in the model enable the implementation of applications of ArcIMap in various educational and practical contexts. An application of ArcIMap must be rooted in its use context, therefore, a study of the social and work processes of the users and the organizational structure of the context in which it will be used must be studied in the design stage of the application. Environments to be used in an educational context have different requirements than ones to be used in practice, because experienced designers have different needs than novices.

Four prototype applications of ArcIMap have been developed, situated, and evaluated in different architectural education and practice contexts. These applications and their evaluation have provided valuable feedback to the theory forming and to the iterative definition of ArcIMap.

The first application is an analysis presentation tool that uses three Ottoman mosques as its case study, researches and validates the unified representational framework and evaluates the notions of interaction and associative browsing in an application of ArcIMap. The second application, Blob Inventory Project (BLIP), is a precedent library designed for modeling knowledge that has emerged from digital design, engineering and production processes of free-form geometry buildings and has been used in the 3rd semester of the M Sc. architecture education. BLIP researches and evaluates user interface and interaction aspects of ArcIMap. The third application, Design Analysis Network (DAN), is an information system implemented as an educational architectural analysis environment used in the undergraduate 2nd year design studio. DAN researches and evaluates all components of ArcIMap, but especially the embedding in a context. The fourth application, DesignMap, is a flexible and extensible content management system intended to be used at the early stages of design, is targeted towards small and medium-sized architectural offices, and has been used and evaluated at the architectural office Mecanoo in Delft. DesignMap researches, implements and evaluates ArcIMap within the context of architectural practice.

The utilizable outputs of this research are the ArcIMap method and model, and the four prototype applications. Additionally, KeySet, as it has been developed and used in DAN, is being successfully used by thousands of students at the Faculty of Architecture, TU Delft, and at the Faculty of Social Sciences, University of Utrecht since 2003, demonstrating the success of the research.

ACKNOWLEDGEMENTS

I have changed countries and continents a few times because of my studies and academic career. The road to this dissertation, the end result of my Ph.D., has also been a winding one.

I would like to express my gratitude to a number of people without whose support this work would have been impossible. First of all, I would like to express my deep gratitude to my promoter Prof. Sevil Sariyildiz for continuously supporting me during the whole time of my employment at Delft University of Technology. She has been a role model for me. Dr. Jan Kooistra deserves a very special credit because of his valuable support in both developing the research ideas and during the final stages of writing the dissertation. Our discussions have been invaluable to me. My gratitude also goes out to my Ph.D. committee members Prof. Gerhard Schmitt and Prof. Ramesh Krishnamurti who have supported me and provided their time and guidance for the conception and continuation of this research. My other committee members, Prof. Chimay Anumba, Prof. Bauke de Vries, Prof. Joop Paul, and Dr. Hielkje Zijlstra have not only reviewed the draft version of this dissertation, but they have also reserved time during the process of this research to personally discuss the points and ideas raised in the dissertation. I also thank my friend and colleague Şaban Özsariyildiz for his valuable comments and discussions throughout this research.

The prototype applications developed, utilized and evaluated as part of this research could not have been possible without the contribution of a number of people. My heartfelt thanks go to Tuba Kocatürk and Martijn Veltkamp for their collaboration in BLIP. My dear friend Tuba has also been a wonderful companion in sharing the ups and downs during the Ph.D. process. Special thanks to Ernst Janssen Groesbeek who welcomingly enabled the embedding of DAN in the BS.C. education. Herman Prast, Steven Steenbruggen and Herbert van Hoogdalem have provided time and support for the use of the DAN application in their design studio. I have sincerely enjoyed our collaboration. Special thanks also to Henry Kiksen and Rubiën Grootfaam who implemented parts of DAN. My gratitude also goed out to Evren Akar and Jelle Attema from the WIT-lab who performed the evaluation of DAN. Additional thanks to my dear friend Evren with whom we had long intellectual discussions and intensely collaborated. Joost Beintema implemented part of the BLIP application and the DesignMap application, together with Zaid Al-Ars. Very special thanks to both of them. Many thanks to Nick Marks, who invested time and effort in the testing of DesignMap. Finally, I thank all the students who used my prototypes and participated in the software evaluations.

I thank my family Süer, Tuna, Müge, Murat, Begüm, Ece, Onur, Mehmet, Dilek, Nihat, Süer, and my Belgian family Lode, Thérèse, Anne, Roxanne, Kristel, Pascal, Ynse, Tibo, Inge, Jan, Rubin and Simon for their endless love, patience and encouragement. I also thank my dear friends Mirjam Foti, Luisa Calabrese, Andrew Borgart, and Selim and Gülay Tangerli for their continuous support and companionship.

Last but not least, I cannot even find the words to thank my dear husband, companion, mentor, and much more, Rudi Stouffs, for always being there for me. This work would have been beyond doubt impossible to achieve without his support, encouragement and patience.

CONTENTS

LIST OF FIGURES AND TABLES

1

INTRODUCTION

Architectural design is a complex process that entails a lot of input and preconceptions. Design problems are primarily considered ill-defined and ill-structured, or even "wicked" (Churchman, 1967; Simon, 1969; Rittel, 1973), and designing cannot be considered as a fully rational process. Additionally, design is an open ended endeavor, because a design is never completed and can always be further improved (De Vries and Wagter, 1990). Understanding and reflecting on design as an activity and process is not straightforward. There have been various studies of design aiming at understanding the acts and phenomena involved in the design process. Generally, studies of design use protocol analysis, observing designers in controlled environments, as a research method, in order to gather data about how designers go about solving design problems and what kind of instruments and heuristics they use to solve these problems (Cross et al., 1996).

Cognitive views of design traditionally consider problem solving activity as an information processing activity and humans to operate as information processing systems (Newell and Simon, 1972). The essence of such a system is its ability to represent events in the external environment and its own operations symbolically, and to manipulate these symbolic representations. In this context, the paradigm of design as a rational problem solving activity that follows this view proposes that design is a process of defining a series of stable problems and respectively searching for design solutions in a solution space (Dorst, 1997). Such a solution space entails information and knowledge related to many aspects of the problem and the solution, and the operations in between. Schön (1983) criticizes the approach to design as primarily a problem solving activity, and argues that considering the design process as ruled by explicit and rational problem solving knowledge is not accurate. He argues that designers know more than they can explicitly tell, which he calls "knowing-in-practice". The design process according to Schön is iterative where the designer alternates between action and reflection on that action, thus "reflection-in-action". Designers use this iterative process to construct meaningful representations of design problems and solutions for themselves.

All of these approaches to design and design research fully acknowledge the need for information gathering throughout the design process, and especially in the conceptual design phase. In the conceptual design phase, there is no ready information about the problem or the solution, and the operations in between are also not known yet. Designers collect, browse, use and reuse, learn from and get inspired by collections of (visual) information especially in the conceptual phase of design. They browse through magazines, scan through or read books, surf the web, and are in general mindful about interpreting what they see within the context of the project they are working on, just waiting to be inspired. The aim of this research is to enable the design and creation of digital environments where a design community can collectively organize and correspond on collections of visual information in a meaningful way. Information repositories already exist in support of later phases in the architectural design process, but needs of designers in the conceptual design phase are not met in such environments where information archival is the main organizational strategy. Visual, flexible and extensible environments are needed in order to support designers in the conceptual phase of design. Such environments are useful both in educational and practical contexts.

In this research, a framework consisting of a method and a (computational) model, and four prototype applications for designers to collect, organize, use, reuse, and correspond on digital design information and knowledge during the conceptual phase of architectural design have been developed.

1.1. RESEARCH MOTIVATION AND GOAL

Design information has many inter-relationships and dependencies, and organizing this information is a complex task. Many digital precedent libraries have been developed, however, most of them do not exceed the functionality of image archives. Documents in such environments are traditionally organized according to categories such as year, style, architect, etc. However, a system that can support designers in the conceptual phase of design in addition to serving as a design information repository is interesting from many viewpoints. This support can be achieved through the encoding, use, and reuse of design information and knowledge and design ideas. The main motivation for such a system is to enable the users to learn from each other (Schön, 1985). Architecture traditionally utilizes the master-apprentice relationship in its instruction strategies, i.e., in the design studio. Therefore, the transfer and reuse of design information and knowledge for further generations is crucial. In a digital precedent repository, the knowledge that resides in the documents stored in the repository, together with the own experiences of its users will provide design support for current and future generations. Such applications targeted for conceptual design that build up dynamic, flexible, extensible, and easy to use knowledge structures do not yet exist in architectural education and practice.

Since the terms information and knowledge are used extensively in this dissertation, it is appropriate to provide a definition for these (Ackoff, 1989). Information is data that has been given meaning by way of relational connection, where data consists of raw symbols. In other words, information is data with some added semantic structure. Information can provide answers to ‘who’, ‘what’, ‘where’ and ‘when’ questions. Knowledge is information plus beliefs, commitments, assumptions, or application. In other words, knowledge is information with some additional semantic and pragmatic structure where information becomes timely, concise and task specific, and therefore useful. Knowledge can provide answers to ‘how’ questions. There are various types of knowledge that can be extracted from documents residing in a digital repository, and from designers using the repository, and these types of knowledge are generally categorized as *explicit* and *tacit knowledge* (Nonaka and Takeuchi, 1995) (Table 1.1). Tacit knowledge is highly personal, context-specific, and therefore hard to formalize and communicate (Polanyi, 1966). Tacit knowledge is knowledge housed in the human brain, such as expertise, understanding,

Table 1.1. Types of knowledge.

Explicit knowledge	Tacit knowledge
codified knowledge that is transmittable in formal, systematic language **Declarative knowledge** seen most often in early stages of learning found in books, research papers, etc. describes the relationships of things in a didactic form **Procedural knowledge** formed later in the learning process describes how designers apply and use their declarative knowledge skill and expertise in design relies on the proceduralization of domain knowledge	highly personal context-specific hard to formalize and communicate housed in the human brain, such as expertise, understanding, or professional insight formed as a result of experience

hunches, or professional insights formed as a result of experience, and can be communicated only indirectly, through metaphor and analogy. Explicit knowledge, on the other hand, refers to codified knowledge that is transmittable in formal, systematic language. Explicit knowledge can also be found in various forms, such as *declarative* and *procedural knowledge*. Declarative knowledge is found in books, research papers, etc., and describes "the relationships of things in a didactic form" (Akin and Akin, 1996). This kind of knowledge is seen most often in early stages of learning. Later in the learning process, procedural knowledge is formed. Procedural knowledge describes how designers apply and use their declarative knowledge.

Skill and expertise in design relies on the proceduralization of domain knowledge (Popovic, 2002). "It is conjectured that learning of skills is contingent on the emergence of procedural knowledge. In fact, in advanced stages where the learner has become an expert, recalling the declarative roots of procedural knowledge becomes increasingly difficult" (Akin and Akin, 1996). It is the purpose of this research to enable the acquisition, representation, sharing and reuse of domain knowledge, and the explication of procedural knowledge.

On this note, it is important to mention that a digital environment for a group of designers to collect, organize, use, reuse, and correspond on digital design information and knowledge during the conceptual phase of architectural design has a number of conditions. One of these conditions is that a digital design information environment must take into account and adapt to the context in which this application will be embedded. Environments to be used in an educational context have different requirements than ones to be used in practice, because experienced designers have different needs than novices (Lawson, 2004; Cross and Cross, 1998; Kaplan et al., 1986; Cross, 2004).

As Archer (1979) states, “there exists a designerly way of thinking and communicating that is both different from scientific and scholarly ways of thinking and communicating, and as powerful as scientific and scholarly methods of enquiry when applied to its own kinds of problems”. Designers work, think and communicate in ways that need to be understood by the designers (and developers) of digital applications that are intended to be used as aids for design. This research does not advocate an encapsulation of a design methodology and offering design aids in steps of this methodology. The proposal, rather, is to provide designers with a general framework that defines the confines in which designers, as a group, think and communicate, and help to construct and visualize their information processes. These information processes are inherently of a social nature and are both the means and result of social interactions among members of a group of designers. Therefore, another condition for a digital design information environment is to enable and support information processes of a group of designers. Such processes are possible within an open system where tools and mechanisms ensure the robustness and continuity of information processes, and where the content is allowed to change and evolve according to the changes in conceptions and cultures. As such, any kind of static ‘design taxonomies’ or other standardization approaches are taken with a pinch of salt.

1.2. RESEARCH QUESTION

It was stated in the previous section that the creation of digital environments for a group of designers to collect, organize, use, reuse, and correspond on digital design information and knowledge during the conceptual phase of architectural design is a goal of this research. In light of the research motivation and this goal, the research question has been formulated in an iterative process as: How can communities of architectural practice correspond on design information and knowledge during the conceptual phase of design?

Communities of practice are formed by people who interact and learn collectively in a shared domain (Bowker and Star, 1999: 294). Students taking the same class and doing group work or architects working at an architecture firm working on a project together form communities of practice. Members of communities of practice generate and handle common knowledge both by reifying, i.e., recording generated and common knowledge into documents in order to support cooperation and mutual understanding of informal group activities among members, and by actively participating in social processes in order to personally contextualize this recorded knowledge (Wenger, 1998). These two activities need to coexist and be combined in order to yield success.

Correspondence is communication with the aim of reaching an agreement on the value of information. Correspondence occurs when members of a community reify and participate in social and information processes of searching and generating, communicating and storing, distributing and exchanging, and validating and discarding information, in a cyclical manner. Through correspondence, information gets value. If users agree on the value of a piece of information, it is validated and may be kept. This process is a discourse mediated and directed by the system components, and is itself an open system. Correspondence on design information and knowledge is a necessary characteristic of an open system.

If successful, this open system is a *complex adaptive system* that is both the means and the result of social processes of a community of practice. By corresponding on design information, members of a community of practice give rise to a complex adaptive system, where the structure of the system is non-hierarchical, the interactions are not predefined, and the state of the system is unpredictable.

Then, how can information systems be created for architectural communities of practice?

Designers collect design documents in order to gain knowledge and inspiration in the conceptual phase of design. However, the information retrieval needs of designers in this phase cannot be easily formulated in terms of well defined queries. A repository of design information must present designers with various ways of retrieving information and knowledge, e.g., a known specific document, or documents pertaining to a certain concepts, or just randomly browsing and following suggested links.

In order to support such information retrieval motives, one needs a *complex information structure* that is composed of information entities and their relationships, tagged with certain *design concepts* in order to be easily retrieved. This network of information entities and relationships must be dense enough in order to yield the users of an information system more than just a few predetermined viewing points. The information entities in this information structure are defined by the documents stored in the information system. The relationships between them can be defined by users or automatically deducted by provided mechanisms of the information system.

The design concepts that tag and classify the information entities are *metadata*. Metadata is data about data and describes the entity in some way. Some examples of metadata are title, author, data of creation, and even more relevant in this context, keywords that describe an entity's content. Such keywords, denoting concepts, are collected in a complex information structure and further related through *semantic relationships*. Some examples of such relationships are is-a, has-a, which are hierarchical in nature, or freely defined associative relationships, such as 'reminds me of'. It is important that metadata and semantic relationships allow for subjective information as well. Being able to freely define associative relations between concepts allows users to record personal associations in their knowledge structures, enabling the recording of internalized procedural knowledge, and possibly even tacit knowledge. Concepts and conceptual relationships together form a *semantic structure* in a complex information structure that acts as the organizational backbone of the information system. Elements (concepts and relationships) of this semantic structure are associated with information entities (documents) and describe them. This allows and supports *associative browsing*, where users browse using the underlying associative relationships between information entities. This enables cognitive jumps and unexpected creative discoveries.

Complex information structures form the basis of the computational framework in order to enable architectural communities of practice to correspond on these structures. The goal of this framework, the *Architectural Information Map (ArcIMap)*, is to define a structure for the design and creation of digital applications that support designers in the conceptual phase of design. It is not the intention to develop a global system that can deal with all documents belonging to all kinds of building projects, but to define the representational framework for achieving an integrated information structure of components, relationships and metadata from a collection of design documents and the knowledge that resides in these documents. The framework can then be implemented for different purposes, domains, contexts, or architectural bodies. ArcIMap defines a semantic structure and a document structure. An application of ArcIMap must be rooted in its use context, therefore, a study of the social and work processes of the users and the organizational structure of the context in which it will be used must be studied in the design stage of the application. Additionally, its users must receive instruction on the concepts behind ArcIMap before using the system.

Applications of ArcIMap enable communities of architectural practice to correspond on design information and knowledge during the conceptual phase of design.

1.3. RESEARCH METHODOLOGY

The research methodology followed for the realization of this research is *grounded theory* (Glaser and Strauss, 1967; Martin and Turner, 1986; Turner, 1983). Grounded theory is "an inductive, theory discovery methodology that allows the researcher to develop a theoretical account of the general features of a topic while simultaneously grounding the account in empirical observations or data" (Martin and Turner, 1986). The grounded theory methodology states that through constant evaluation of activities and evaluation results, an information system gets grounded in its context. This is an iterative process of data collection, analysis and interpretation. This iterative process leads to a theory formation.

This inductive approach is in accordance with the viewpoint of design research where empirical observations of designers' activities result in design knowledge. Within the context of this research, the dynamic nature of this approach provides the flexibility and support needed for a descriptive theory to respond to the research question. The perspective of this research is interpretive rather than positivist, and therefore, the fact that the grounded theory methodology brings forward context and process makes it highly suitable for this research. This is a qualitative research relying on the interaction of contextual conditions, processes, social organizations and actors.

The empirical data and observations for the inductive theory formation have been provided by a number of case studies. Structured and unstructured interviews, questionnaires, design workshops, structured brainstorming sessions, analysis of recorded processes, and observation have been utilized. Knowledge from relevant literature has also contributed to the data and analyses in the process of forming a grounded theory. The selection of subjects for the data collection has always been carefully selected in terms of organization and context, not to 'blur' the theory forming.

A number of fields provide input to this research. *Architectural design theory* sets the context of the research as a whole. *Cognition* plays a role in defining a model for information organization that supports the cognitive design thinking of designers. Another field that provides support to the research as a whole is *computer science* and its technologies and applications. *Web technologies* provide support for the conception and implementation of prototype applications. *Knowledge representation*, and in particular *conceptual structures*, provides a framework for knowledge organization that forms an indispensable part of the research result. *Semantics*, a branch of linguistics, provides an understanding of relationships between design entities. *Library sciences* offer insight into the organization of entities in the form of subject-based classification techniques. The *theories and methods of learning* also provide input: *constructivist theories of learning* provide a basis for concept mapping, which is used in this research as a method for building up knowledge structures. *Concept maps*, the products of concept mapping, have their roots in *education and learning sciences*.

Figure 1.1 demonstrates the research process and the research outputs. All the fields mentioned above provide input into the definition of the ArcIMap framework. *Design research* and *computational research* set the context, requirements, and theory for the formation of ArcIMap. There are three main outputs of this research:

- ArcIMap as a computational model
- ArcIMap as a method
- Four prototype applications of ArcIMap

ArcIMap method: ArcIMap is a method that defines social and information processes in order to create complex information structures underlying complex adaptive systems, which are the means and result of these processes. These social and information processes are derived from interactions of communities of practice. ArcIMap encapsulates methods to describe these processes in order to be able to implement applications that are rooted in their context.

ArcIMap model: ArcIMap is a computational model that is represented as an object model. The model is derived using the Object Oriented Software Engineering (OOSE) method (Jacobson et al., 1992) and is represented in UML (Unified Modeling Language). Requirements have led to the object model and the object model is enriched with use cases. The model acts as a structure for the design of information systems. The techniques and technologies encapsulated in the model enable the implementation of applications of ArcIMap in various educational and practical contexts. The model does not prescribe the context, content and users of the implemented applications.

Four prototype applications of ArcIMap for use in architectural education and practice: Four prototype applications that implement, research, and evaluate certain aspects of ArcIMap as a method and model have been developed, used, and evaluated. The first prototype implementation is an analysis presentation tool that investigates the data structure, interaction principles and technical considerations for an information organization environment. The second one is an educational prototype implementation of a cooperative knowledge base that supports the process integration between form-finding, structural design and manufacturing domains in the design of double-curved surface buildings. The third one is an educational prototype implementation of an architectural analysis environment that encourages design correspondence among students and enhances the design understanding of students. The last one is a prototype implementation of a design information organization system for small and middle sized architectural offices that enables correspondence within small project teams and supports the storing of design documents for office-wide information reuse.

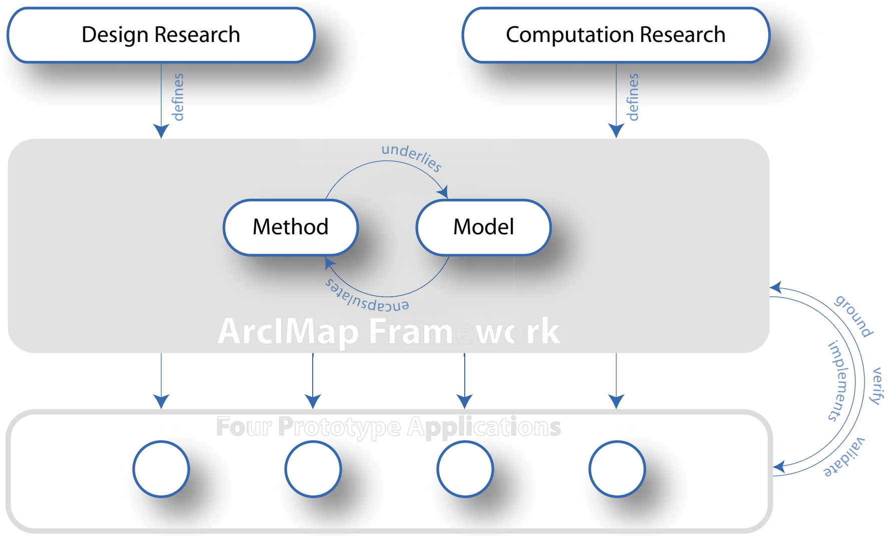


Figure 1.1. The research process and the research outputs. Design research and computation research set the context, requirements, and theory for the formation of the ArcIMap framework. ArcIMap encapsulates methods to describe social and information processes in order to be able to implement applications that are rooted in their context. ArcIMap defines a computational model that acts as a structure for the design of information systems. Within the ArcIMap framework, the method underlies the model and the model encapsulates the method. Four prototype applications are implemented using the framework. These applications ground, verify and validate the framework.

The main societal contribution of this dissertation is that through an implementation of the framework that it proposes, design information and knowledge can be reused, resulting in the improvement of quality, and gain of time, and therefore costs, in the design of architectural projects.

1.4. OVERVIEW OF THE DISSERTATION

After this Introduction chapter, Chapter 2 sets the context for this research. The goal is the definition of a digital design information environment for the conceptual design stage that enables a complex adaptive system. The chapter focuses on the conceptual architectural design phase, and discusses the need of designers for looking at, to learn from, and to be inspired by collections of design documents and precedent libraries in this phase. The encoding and organization of precedent knowledge is discussed in relation to the collective definition of a flexible and extensible organizational structure for this knowledge. This knowledge and information organization occurs as the means and result of social processes within communities of architectural practice where members reify and record common knowledge and understanding into documents, and actively participate in order to personally contextualize this recorded knowledge. These social activities enable correspondence: communication with the aim of reaching an agreement. This results in an open system, a complex adaptive system, in which the members of a community of practice constantly perform the four information process activities in a cyclical manner: searching/generating, communicating/storing, distributing/exchanging, and validating/discarding information and knowledge. As part of this information process users index documents using metadata, associatively browse the information and knowledge residing in the environment, and create complex information structures with a high density and intensity of relationships among information and knowledge entities. Later in this chapter a number of relevant exemplary applications of digital precedent-based design systems and their criticism are presented. The chapter ends with an educational scenario for a complex adaptive system, and looks ahead on the ArcIMap framework as an aspiration.

Chapter 3 describes and discusses complex information structures enabling complex adaptive systems and as the underlying foundation of ArcIMap. These complex information structures consist of documents depicting design information, an organizational structure that is built up by the members of an architectural community of practice giving the members the possibility to reflect on the information and knowledge entities, metadata describing these documents, and a specification of the relationships between these documents (or parts thereof). When building such structures, both modeling and visualizing the complexity require careful consideration. A representational language as a common syntax for the representation of complex information structures is adopted, and two techniques to increase the structure's cardinality and its interrelatedness: the separation of the document structure and semantics, and the decomposition of documents by content are presented. In this chapter, first how complex information structures are built up is described, then the separation of the document structure and the semantics is described and two main components of complex information structures are proposed: the semantic structure and the document structure. The next section describes the semantic structure: it is built up collectively by the users of a system and acts as a backbone for information and knowledge organization, it is built up using concept mapping as a method, it is visualized using dynamic graph (network) visualizations, and it is structured and represented as a semantic network made up of concepts and conceptual relationships where the conceptual relationships are semantic relationships. The document structure consists of a collection of multi-media design documents, where the documents are further decomposed and indexed by elements of the semantic structure according to their content, enabling users to access specific information directly instead of requiring a traversal of the document hierarchy, and allowing alternative views to those that are expressed by the individual documents. This chapter ends with describing technologies for decomposing documents.

Chapter 4 formally describes the ArcIMap framework. The goal of ArcIMap is to define a framework for the design and creation of digital applications that support designers collecting design documents in the conceptual phase of design. It is not the intention to develop a global system that can deal with all documents belonging to all kinds of building projects, but to define the representational model for achieving an integrated information structure of components, relationships and metadata from a collection of design documents and the knowledge that resides in these documents. The framework can then be implemented for different purposes, domains, contexts, or architectural bodies. This chapter first lists some provisions for the definition of the ArcIMap framework followed by an informal description of the two components of the framework: the semantic and document structures. Next, the ArcIMap object model is formally described. In order to demonstrate a real-life use of the model, the description of the educational scenario previously introduced in chapter two as the aspiration for ArcIMap is demonstrated. This scenario has been further adapted to the structure and process defined by ArcIMap. Finally, a number of use cases are included in order to understand and explicate specific needs in the application design from the viewpoint of the users. The chapter ends with a number of guidelines for the design and implementation of an application of ArcIMap.

Chapter 5 presents four prototype applications of ArcIMap, each implementing, testing, researching and evaluating certain aspects of ArcIMap. An *Analysis Presentation Tool* that uses three Ottoman mosques as its case study, researches and validates the framework's unified representation, and evaluates the notions of interaction and associative browsing in an application of ArcIMap. *Blob Inventory Project (BLIP)* is a precedent library designed for modeling knowledge that has emerged from digital design, engineering and production processes of free-form geometry buildings and has been used in the 3rd semester of the M Sc. architecture education. BLIP researches and evaluates user interface and interaction aspects of ArcIMap. *Design Analysis Network (DAN)* is an information system implemented as an educational architectural analysis environment used in the undergraduate 2nd year design studio. DAN researches and evaluates all components of ArcIMap, but especially the embedding in a context. *DesignMap* is a flexible and extensible content management system intended to be used at the early stages of design, is targeted towards small and medium-sized architectural offices, and has been used and evaluated at the architectural office Mecanoo in Delft. DesignMap researches, implements and evaluates ArcIMap within the context of architectural practice.

Chapter 6 states the results and contributions of the research and sets an agenda for further research in this area.

2

INFORMATION AND KNOWLEDGE ORGANIZATION IN CONCEPTUAL ARCHITECTURAL DESIGN

Many design researchers define the design process in terms of design phases. Asimow (1962) defines three design phases: feasibility study, preliminary design, and detailed design[1] (Figure 2.1). In the feasibility study phase the designer investigates the design context and identifies design problems and performs a quick survey of possible and useful alternatives as solutions. The preliminary design phase looks at the set of possible alternative solutions from the feasibility study phase and aims to identify one or only a few preferred solutions for the design. The detailed design phase begins by taking the preferred concepts selected in the preliminary design phase and ends with a complete representation of the design. Asimow's feasibility study phase and preliminary design phase together form the *conceptual design phase*.

The goal of the conceptual design phase is generally considered for the designer to come up with one or a number of design concepts that will prove to be powerful and that will 'hold' throughout the rest of the design process (Heylighen and Neuckermans, 2000). The designer creates a new artifact of which the properties are at best only partially known, and strives to define both the components of the designed object and the relationships between them (Goldschmidt, 1994). In the conceptual design phase, a large number of constraints and inputs play a role, and these criteria come from a very large range of contexts. Designers consider, among others, the physical context, the function, owner and user requirements, available resources, and other constraints imposed upon the design (Murthy and Lutton, 1994).

In the conceptual design phase, there is no ready information about the problem or the solution, and the operations in between are also not known yet. Designers perform *problem setting*, which is "a process in which, interactively, we name the things to which we will attend and frame the context in which we will attend to them" (Schön, 1983: 40). Designers frame a design problem, "set its boundaries, select particular things and relations for attention, and impose on the situation a coherence that guides subsequent moves" (Schön, 1988: 182). During the problem setting activity, the situation is compared with the designer's prior knowledge, experience and strategies, and a frame of reference is created for the designer to modify existing strategies or construct new strategies.

[1] These design phases are ordered chronologically according to the sequence of a design project. These phases, however, are not prescriptive in the sense that they must be rigidly followed in each project; phases may overlap or be omitted depending on the particular project.

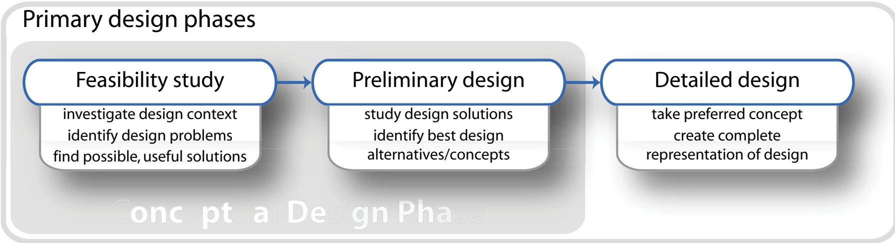


Figure 2.1. Phases of architectural design are feasibility study, preliminary design, and detailed design (Asimow, 1962). Feasibility study and preliminary design together form the conceptual design phase.

Designers therefore collect information on and study reputable design solutions and strategies. The designer gains experience when she learns by adding a new strategy to her repertoire. In this process, designers need to draw upon external information in order to compensate for missing information and knowledge, in accordance with the level of their design expertise, and use this to construct the problem space (Simon, 1973). After all, the conceptual design phase is the most critical phase of all. The most, and additionally, the most influential design decisions are made in the conceptual design phase (Sariyildiz, 1991). Therefore, designers tend to collect and research large amounts of information in the conceptual design phase. They visit buildings, browse books and magazines, look at digital collections on the internet, and are generally aware and mindful throughout the day with the design context in the back of their heads. This information also serves as a source for inspiration and an enabler for creative coincidences for designers (Murty and Purcell, 2002; Eckert and Stacey, 2000; Goldschmidt, 1994; Gross and Do, 1995).

The focus of this chapter is design research and cognition: the encoding and organization of precedent knowledge is discussed in relation to the collective definition of a flexible and extensible organizational structure for this knowledge. This knowledge and information organization is discussed within the context of architectural communities of practice, where members reify and record common knowledge and understanding into documents, and actively participate in order to personally contextualize this recorded knowledge. A cyclical information process of searching and generating, communicating and storing, distributing and exchanging, and validating and discarding information and knowledge within the community of practice leads to a complex adaptive system. Members of this system index documents using metadata, associatively browse the information and knowledge residing in the environment, and create complex information structures with a high density and intensity of relationships among information and knowledge entities. Later in this chapter a number of relevant exemplary applications of digital precedent-based design systems and their criticism are presented. The chapter ends with an educational scenario for a complex adaptive system, and looks ahead on the Architectural Information Map (ArcIMap) framework as an aspiration.

2.1. VISUAL ANALOGY AND METAPHOR

Since the designer knows, thinks and works in visual ways (Cross, 1982), collections of visual information are often used for conceptual design. Concepts in architectural design are often more easily and completely expressed by visual information – images, pictures, diagrams, drawings, etc. – than in words. Visual information entities are usually documents depicting other designs, related to the design task at hand, but sometimes they are from completely other contexts. Designers search for sources of shapes, forms, patterns, materials, etc., that can be translated into aspects of designs at hand (Eckert and Stacey, 2000). In this translation process, a ready made external representation can show the designer a way to explore and map solutions to the design at hand. Designers use visual analogy in order to translate and map information and knowledge from these external references into their own design (Goldschmidt, 1994; Akin, 1989; Broadbent, 1988; Lang, 1987; Rowe, 1987; Steadman, 1979) (e.g., Figure 2 2). Goldschmidt (1994) calls this process "transferring a diagram from a source structure to a target domain", and claims that even though this transfer may be incomplete, it "enables the organization of components of the target structure into a coherent system." Analogies can be deduced between objects in the same domain, or objects from completely different domains.

The process of using analogies in design exploration is called *analogical reasoning*. Analogical reasoning is "based on the idea that problems or experiences outside the one we are currently dealing with may provide some insight or assistance" (Maher et al., 1995: 1). Analogical reasoning plays a crucial role in design (Gross and Do, 1995; Casakin and Goldschmidt, 1999; Falkenhainer et al., 1989; Bhatta et al., 1994; Casakin and Goldschmidt, 2000; Qian and Gero, 1992; Casakin, 2004; Gero and Maher, 1992; Tzonis, 1990; Chiu and Shih, 1997), because designers frequently utilize analogical reasoning during concept generation when looking at visual information collections, getting insights and assistance from established design solutions and design strategies.

An analogy in design is regarded either as a surface analogy or a structural analogy. A surface analogy "relates to easily accessible or superficial concepts of object properties", whereas a structural analogy "involves a system of higher order relations that are based on deep properties of a familiar situation" (Casakin, 2004). In order to use structural analogy, a designer must be aware of and extract knowledge from the source object in order to apply it in the target object. The use of structural analogy requires a deeper understanding of the inner structure of the source. The use of metaphor goes hand in hand with the use of structural analogy in architectural design (Lawson and Loke, 1997). Analogy has to do with similarity, whereas metaphor is the transfer of a concept to another context (Schön, 1993). Analogy concentrates on visual similarity between the source and target objects, whereas metaphor concentrates on semantic relations (Casakin, 1997: 16). Designers use analogies and metaphors while reasoning on design issues without consciously differentiating between the two, and tend to communicate ideas, share their knowledge and experiences with others using sketches, diagrams, narratives and stories full of analogies and metaphors (Herschel et al., 2001; Woo et al., 2002; Turner and Turner, 2003; Lloyd, 2000; Lawson, 2004; Oxman and Oxman, 1993; McDonnell et al., 2004). Analogy and metaphor are useful instruments for expressing tacit knowledge as explicit knowledge (Nonaka and Takeuchi, 1995: 12-13). Using metaphor and analogy is "a way for individuals grounded in different contexts and with different experiences to understand something intuitively through the use of imagination and symbols. No analysis or generalization is needed"

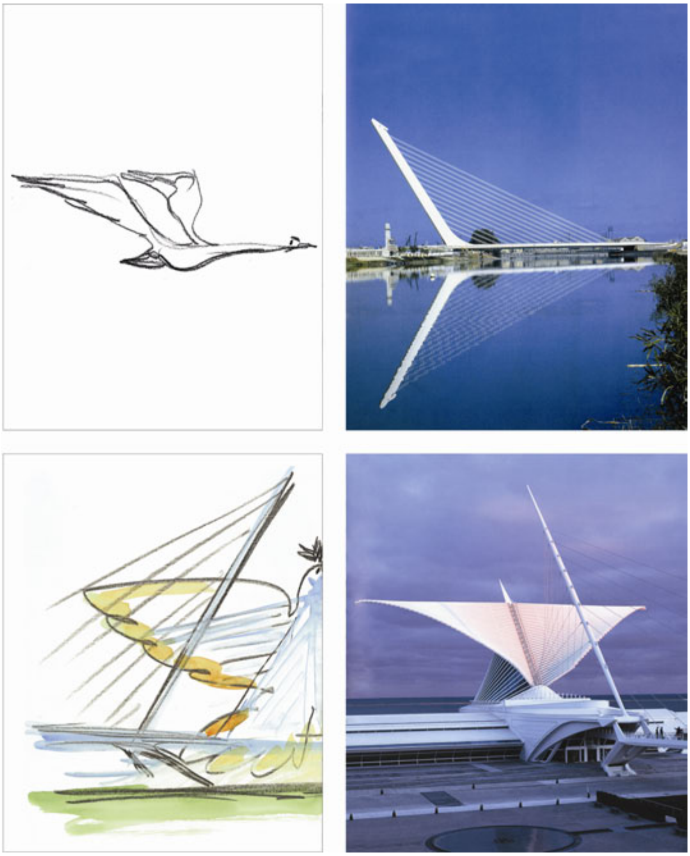


Figure 2.2. An example of the use of visual analogy in design by Santiago Calatrava. Top: Alamillo Bridge; Bottom: Milwaukee Art Museum. (Images from: Top left Blaser, 1989: 157; the rest Tzonis, 2004: 147, 294, 291)

(Nonaka and Takeuchi, 1995: 13). Metaphors enable designers to begin to express what they know but cannot yet say, making them highly suitable for early stages of design and knowledge creation. An analogy compares two ideas and objects distinguishing how they are alike and not alike, and is therefore an intermediate step between pure imagination and local thinking. In this context, analogy and metaphor together foster knowledge creation and recording in the conceptual design phase.

2.2. PRECEDENTS AND DESIGN KNOWLEDGE

Precedents are "specific designs or buildings, which are exemplary in some sense, so that what architects and students glean from these examples can support their own designs. These precedents are very often past solutions to specific design problems" (Akin, 2002). Precedents contain design knowledge that can be accessed and reused in the context of the design problem at hand. The knowledge that precedents provide to designers is both physical and conceptual. The conceptual knowledge in precedents often provides the underlying order and structure of an aspect of the design. This underlying order and structure integrate the physical elements of the building design, according to various functional, formal and esthetic values, based on some 'design concepts'. Students usually have analysis tasks in design studios in order to understand and extract precedent knowledge from documents – plans, sections, elevations, pictures, etc. – of precedent buildings.

Precedent-based design is the selection of relevant ideas from prior designs in current design situations. *Precedent-based learning* is the common method used in the architectural education system. In order not to reinvent the wheel over and over again, we learn from our elders and adopt their successful solutions to situations similar to the ones we cope with (Goldschmidt, 1995). Design studio instruction aims at helping students learn to understand design principles, formulate, isolate and define design problems, and to use this knowledge in their design. Robust and context independent principles are hardly present in this instruction. Instead, students are given many precedents with the purpose of learning many heuristics from them. This method of instruction is experiential and the knowledge gained by the students is situated.

Precedents have been the subject of numerous design research related investigations (Tzonis and White, 1994; Oxman and Oxman, 1994; Kolodner, 1993; Maher et al., 1995; Heylighen, 2000; Schmitt, 1994; Fang, 1993). Many studies focus on how precedent knowledge influences design concepts. It is commonly accepted that in order to reason with precedents, one needs already existing domain knowledge and experience. It is easier for experienced architects than novices to deduce knowledge from precedents and to use this knowledge in designing. When experienced designers view a precedent document, they can quickly recognize if that document is relevant and useful for the design project and context at hand. This has been explained in the literature by the fact that experienced designers perceptually recognize knowledge rather than analytically study the design: experienced designers recognize the 'schemata' underlying the precedent, i.e., the patterns that form the conceptual design variables that organize the precedent. "The more experienced a designer, the more likely it is that perception of drawings will be by recognition of schemata that conceptually organize [the] precedent" (Lawson, 2004: 451). Design students or novice designers, on the other hand, do not have a mental and physical repository of schemata. Expert designers can remember solutions posed by known and studied design precedents. They have, in the course of their design career, worked on similar examples and contexts, and have developed similar design solutions.

The next step for forming valuable design knowledge after being able to recognize schemata in a design precedent is reaching an understanding of the values and properties that are achieved by using a certain scheme in a project. The term *guiding principles* has been used to denote principles in a project that are driving forces behind a certain design concept (Lawson, 2004). For example, the main guiding principle in Calatrava's designs is movement (Tzonis, 1999). Other guiding principles could be geometry, light, or movement patterns in space. Expert designers have and use established guiding principles. By studying such designs, these guiding principles can be detected, and furthermore, sometimes, the precedents that have inspired these can be detected. The detected principles are physical and/or conceptual.

Design students need to gain the experience to be able to easily extract design knowledge from design precedents in order to isolate problems and issues in conceptual design. Therefore, in support of educating designers, a collection of relevant design precedents and an organization of the structure of the knowledge they contain is indispensable in a design education context, as well as in offices for novice designers to learn from experienced designers. A digital precedent information organization environment is a suitable instrument for this purpose.

Representations are crucial in design problem solving (Akin, 2001). Designers communicate design issues and knowledge to themselves and to others though design representations. Precedent documents are represented in many different formats, and convey the viewpoint of their creator and contain knowledge about one or more specific issues of design. These documents are used by the individual designers as well as other team members. Collecting and organizing the precedent information in a way that enables the communication of the embedded knowledge in the precedents is highly beneficial. In order to support architectural design students or practitioners in the conceptual phase of design, mechanisms that support the building up of precedent libraries are necessary. A system that can be used to store, organize, and present precedent information and knowledge is indispensable. Additionally, mechanisms specifically for the acquisition and communication of conceptual and physical precedent knowledge are needed.

In a precedent information organization environment, the amount and variety of information is important for its usability. This environment is used by designers to get ideas and insights, encounter creative coincidences, and gain knowledge. However, the assumption that the more knowledge a designer has the more design skill the designer acquires is not true (Oxman, 2004). Design competence has to do with knowing where to find the knowledge and knowing which specific knowledge to apply in a specific situation and how to apply it (Cross, 1982). Therefore, in a precedent library, the quantity of information is important, but not the most important aspect. “Educational research suggests that the organizational structure of knowledge is at least as important as the amount of knowledge in understanding any particular knowledge domain” (Baron and Steinberg, 1987). In order to support designers in an electronic environment, a framework for the information organization is needed that allows one to structure knowledge, individually or in group. This structure should support designers to encode their explicit design concepts as well as their subjective ideas in order to support the forming of a common language. In an electronic precedent environment, the ability to encode, search and extract design knowledge relevant to the problem at hand is highly significant (Oxman and Oxman, 1993). The organizational structure should reflect a cognitive model of design reasoning in order to support design competence and the retrieval of knowledge during design. “A cognitive model of design provides for relevant connections to be made between the current problem and the design rationale and concepts underlying precedents” (Oxman and Oxman, 1993). The assumption here is that by allowing the unrestricted encoding of personal notions and insights, the acquisition of procedural internalized knowledge will be enabled to a certain extent. Flexibility and personalization play an important role in achieving this. The organizational structure is the foremost instrument in an electronic environment that enables the acquisition, use, and reuse of precedent knowledge. When this extraction and structuring of knowledge is done by the users of the system rather than its developers, it gives the users the possibility to reflect on the knowledge, and to consciously think about the knowledge entities and their relationships.

2.3. COMMUNITIES OF PRACTICE AND COMPLEX ADAPTIVE SYSTEMS

Designers communicate using countless analogy and metaphors, in the form of narratives and stories, and during such design conversations, use words and drawings (pictures, diagrams) to convey the embedded knowledge in these conversations. In order to be able to understand the numerous references in design conversations, one needs to possess certain (domain) knowledge and a common understanding of a vocabulary shared by a (professional) group, i.e., the design community. Hence, groups of design professionals share a *common language* (Lawson, 2005; Lloyd, 2000). This common language also contains vocabulary derived from the common use of analogy and metaphor.

An example of communication in a design office using vocabulary from a common language is provided by Lawson (2004):

> *"Listening to conversation in such practices reveals just how extraordinarily efficient communication becomes since enormously complex and sophisticated sets of ideas can be referred to using simple diagrams, catchphrases (for example, 'round shapes in square containers') or even single words (for example, 'belvedere'). Such a phenomenon is hardly new to architects or designers in general. It is precisely that of concept formation or the development of schemata. For experienced architects, the concept or schema of 'round shapes in square containers' includes not just the simple idea of geometry but the whole game of contrasting curved and straight lines, and probably many related precedents. … They collectively delight in these ideas and have studied them and exploited them in previous designs."*

Since a group of designers use a common language for communication, the vocabulary of this language refers to knowledge shared by the members of the group, and designers reuse their own designs and design experiences, an environment that stores and organizes precedent information and knowledge should also support the recording, forming and maintenance of common knowledge among its users in order to support novice designers to connect terms to schemata that can be understood within the community. The members of the community, designers, students and professionals, do not only learn from precedents, they also learn from and with each other (Schön, 1985: 6), through reviews and critics, but also through direct communication and correspondence on each other's work (Stouffs et al., 2004a). Such a community within the context of architectural designers – practitioners or students – can be called a community of practice.

"Communities of practice are groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly"[2] (Wenger, 1998: 72-85). People who learn collectively in a shared domain form communities of practice (Bowker and Star, 1999: 294). Learning in a community of practice is not limited to novices, everyone in the community learns. Communities of practice have existed as long as humans have learned together, and humans belong to numerous communities of practice

[2] Community of practice as a term has been introduced by Jean Lave and Etienne Wenger as a learning model (Lave and Wenger, 1991). The concept has its roots in learning theory, but is also used in various fields including education, sociolinguistics, anthropology, and knowledge management.

throughout their lives. The combination and parallel development of three important elements cultivates a community of practice (Wenger, 1998):

- Members share a domain of interest and are committed to the domain, and therefore possess a shared competence that distinguishes them from other people.
- Members interact and engage in joint activities and discussions in order to learn from each other and share information within their domain.
- Members are practitioners who develop a shared repertoire of resources: experiences, stories, tools, and ways of addressing recurring problems – in short a shared practice. This takes time and sustained interaction.

Communities of practice perform, among others, activities for problem solving, information and experience sharing and reusing, and documenting and mapping knowledge. Members of communities of practice operate in terms of generating and handling common knowledge both by recording knowledge into documents in order to support cooperation and mutual understanding of informal group activities among members, and by getting actively involved in social processes in order to personally contextualize this recorded knowledge. It is the combination of these two activities that enables knowledge handling within communities of practice. In this process of knowledge handling, explicit and tacit knowledge and formal and informal processes are not separated. Wenger (1998) calls this process of knowledge creation and handling in its entirety "reification" and "participation", Kooistra and Hopstaken (2002) call it "ice-canoe".

Creating a digital system for acquiring and managing precedent knowledge collectively as a community of practice has many advantages:

- Members take collective responsibility for managing the knowledge they need, recognizing that, given the proper structure, they are in the best position to do this.
- Communities among practitioners create a direct link between learning and performance, because the same people participate in communities of practice as in design teams.
- Members can address the tacit and dynamic aspects of knowledge creation and sharing, as well as the more explicit aspects. (Wenger, 1998)

In the context of an electronic environment that enables the acquisition, use, and reuse of precedent knowledge where the knowledge is built up and used by members of a community of practice, tools and mechanisms for meaningful transactions and correspondence among members are needed in order to create a useful and usable environment.

Correspondence is communication with the aim of reaching an agreement. A digital environment possessing the qualities described above can play a role in this, offering designers access to a broader selection of work and offering additional means for communication and correspondence. As design practice increasingly becomes distributed and multi-disciplinary, "in a design practice [design] knowledge has to become common or shared for the team to operate effectively" (Lawson, 2004: 453). A system where a design team can simultaneously and collaboratively model their cognitive design models is important as a design aid, but it is also important as a means to record design knowledge and experiences for future generations.

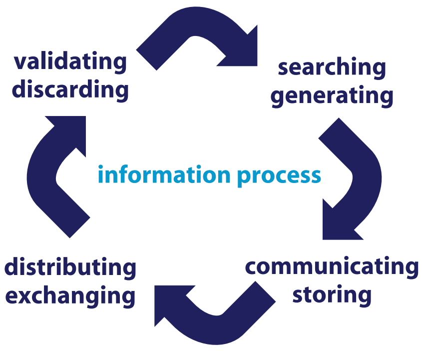


Figure 2.3. The information process adopted for correspondence in a community of architectural practice. This process considers four main groups of information activities in a cyclical manner.

Such an environment relies on the specification of an information process that defines the activities related to information and knowledge on a general level. The information process adopted in the InfoBase project[3] (Kooistra et al., 2003) is suitable for this research. Its purpose in InfoBase is "to support the students in their switch to the professional world through means of locking scientific value into the students' communication processes among themselves, with their instructors and also with researchers" (Stouffs et al., 2004a). It can provide the same support in a practical context, with respect to the communication in a design team, and with respect to the communication of novice and expert designers in a professional context.

The adopted information process considers four main groups of information activities in a cyclic process: searching and generating, communicating and storing, distributing and exchanging, and validating and discarding information and knowledge (Kooistra and Hopstaken, 2002) (Figure 2.3). The goal is to achieve a system that acts as a "deputy: a partly virtual and partly human substitute that has the (managing) power to deal with collected information, to make conclusions, to take initiatives" (Kooistra and Hopstaken, 2002). This 'deputy' can be a rigid one, where the information or knowledge that is stored in the system is fixed: this is a closed system. A discourse of humans, mediated and directed by the system components, is an open system. Open systems according to Popper (1982: 173) are systems in a state, far from equilibrium, that show no tendency towards an increase in disorder. This structural disorder in the system is ensured by the nature of the open system when it is considered in terms of a dynamic set of interacting entities where

[3] The aim of InfoBase is to teach students how to deal with metadata for the exchange of knowledge and information within a professional community or network of students, educators and researchers. Four aspects are distinguished within this aim: 1) [research wise] we a scientific approach is considered for dealing with information and communication by means of the application of metadata; 2) [didactically] corresponding to this approach, method for adding, using and managing metadata is developed and this method is implemented in a learning path with increasing responsibility for the student; 3) [technically] a digital environment, named InfoBase, is developed to support this learning path and to support students to store, exchange and manage the information they collect and generate by means of metadata, both individually and in group, and independently; 4) [strategically] students are encouraged through this system to learn from one another and to collaborate as young professionals by directly comparing their work and designs; at the same time, the student is introduced to the paradox that a database loses information quality when it is cleaned up into a tidy database. http://infobase.bk.tudelft.nl

no single individual or organization is in control of the construction or, consequently, behavior of the set as a whole (Frederiksson and Gustavsson, 2001). An open system is needed in order to ensure reification and participation in a community of practice, because a community of practice is non-hierarchical and dynamic, and its interactions are not predefined.

We can consider such an open system as a *complex adaptive system*. According to Dooley (1997), "a complex adaptive system behaves/evolves according to two key principles: order is emergent as opposed to predetermined, and the state of the system is irreversible and often unpredictable." Examples of complex adaptive systems include social insect colonies, the brain, traffic jams, national economies and stock markets, industrial infrastructures, and any human social group-based endeavour in a cultural and social system such as communities of practice (Yang and Shan, 2008). Emergence and self organization are important for the formation of complex adaptive systems.

When the content of a precedent knowledge system is both means and result of social processes of a community of practice, in the form of correspondence, it can be said to support a complex adaptive system. In order to do so, it must be both robust and flexible (Stouffs et al., 2004a). Robustness in this context means that the system must offer mechanisms, that don't change over time, for the participants to be able to correspond on the content. Flexibility in this context is allowing the content to change and reflect (on) the changes and evolutions that come with the state-of-the-art of technology, society and culture.

Correspondence was described earlier in this section as communication with the purpose of achieving agreement. In the context of an information system, the agreement is on the value of information. Through correspondence, information gets value. If users agree on the value of a piece of information, it is validated and can be kept. Correspondence on design information and knowledge is a necessary characteristic of an open system.

In the process of document organization, the issue of the value of the document, and therefore the value of information contained in the document, is a critical factor in deciding both to keep the document, and also how it should be placed within an organization scheme. This is especially important when there are a very large number of documents available. A document has value if it is of actual or potential use to someone. The most important factor in determining the value of information is figuring out its importance and why it is important (Megill and Schantz, 1999). Through correspondence, this is done collectively by the users of the system. The process of correspondence is achieved when a user creates a document in the system and she makes a number of claims about this document. Other users may agree with these claims, either explicitly, or simply by using or reusing this document. If the document is unused or the claims not agreed upon, the document becomes redundant and obsolete in time. The value of a piece of information is a qualitative judgment, not a quantitative one. The value is not static either; it changes over time.

In the case of archiving documents, determining which documents should be kept and how these should be indexed is the responsibility of the document manager. Archiving originates from the library sciences and is applied to collections. One or more archivists are designated that organize the documents produced and used within an organization. A document in a collection is considered as an object that can be catalogued (Megill and Schantz, 1999: 19). This is a rather static approach.

Instead, when the purpose is not to have a designated archiver, but when the creators of the documents are the same as the organizers and the users, the information system is alive. In such a system, frameworks are established to collectively manage the corporate memory without having archivers making judgments on the value of each individual item.

2.4. INFORMATION PROCESSES IN COMMUNITIES OF PRACTICE

As discussed in the previous section, communities of practice perform activities for problem solving, information and experience sharing and reusing, and documenting and mapping knowledge. Members of communities of practice operate in terms of generating and handling common knowledge both by recording knowledge into documents in order to support cooperation and mutual understanding of informal group activities among members, i.e., reification, and by getting actively involved in social processes in order to personally contextualize this recorded knowledge, i.e., participation. Procedural and technical support is needed for reification and participation within an architectural community of practice within the context of conceptual design.

Reification in this context has to do with the collection of (precedent) information. Architectural objects are generally expressed through abstractions[4]. A geometric model specifies a single abstraction; other abstractions express other aspects of the object, such as function, acoustics, structure, process, form generation, space, and organizational relationships (Schmitt, 1993: 39) (Figure 2.4). Abstractions may be described in different formats such as drawings, diagrams, models, pictures, and textual information, and are individually contained in different documents. These documents accumulate over time in a shared space.

In order to activate collaborative learning, a number of activities within an information process are considered: indexing information using metadata, retrieving information, and associatively browsing information.

2.4.1. Metadata, classification, and document indexing

As discussed before, the process of correspondence is initiated when a user creates a document in the system and she makes a number of claims about this document. Correspondence is achieved when others in the system agree (or disagree) on these claims. An explicit way of making claims on a document is tagging this document using metadata.

Metadata[5] is generally described as 'data about data'. Metadata is information that describes an object or a document in any way. The way the metadata is represented is independent of its content. Examples of metadata for a document are its title, author,

[4] The term "abstraction" concerns an abstraction of the content of a design object, not the physical electronic form of the document.

[5] There is a standardized set of metadata defined in the Dublin Core that aims at standardizing the general metadata for effective information exchange (http://dublincore.org/).

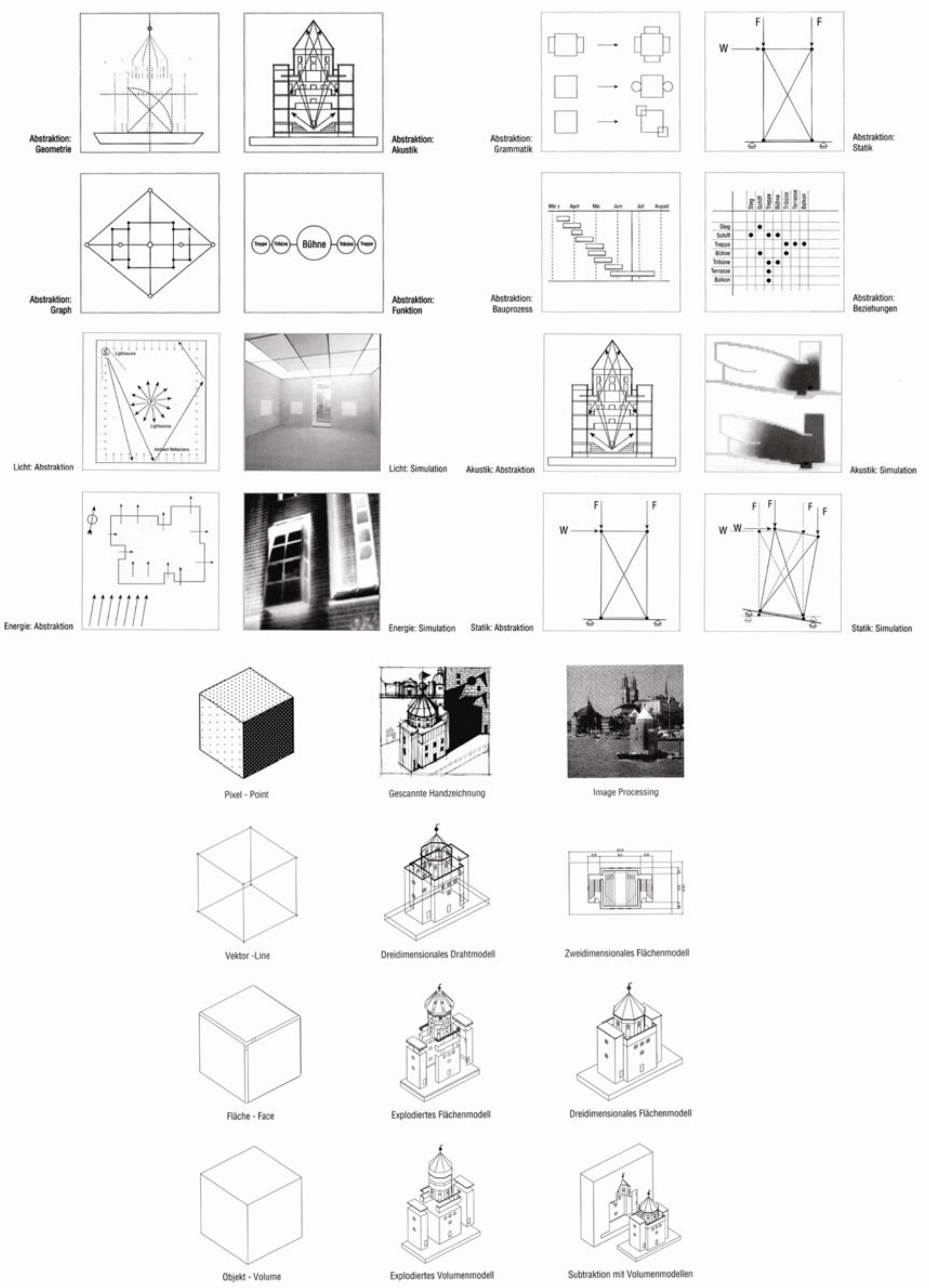


Figure 2.4. A demonstration of a collection of abstractions. These abstractions define various aspects of the buildings they depict. Diagram after Schmitt (1993).

access rights, access information, keywords, marks, ratings, etc. One of the ways in which metadata describes documents is by connecting documents to the subjects they are about (Garshol, 2004: 381). When the metadata is created by members of a community of practice in the form of keywords, reification and participation are at play, because members of the community are making claims about information and corresponding about these claims. Such metadata also immediately places a document in an organizational structure (and creates an organizational structure), by indexing this document with this metadata.

From the viewpoint of information organization, documents are indexed[6] such that each document is represented by a set of metadata. This indexing can be done manually or automatically. Metadata can be generated automatically from the content of a (textual) document by using text mining techniques next to allowing users to freely define additional keywords that describe the content of the document. Metadata derived from the content of the information residing in the system is especially useful in information organization and retrieval, because this metadata actually says something about what the information is about. In general, information retrieval in such environments is based on searching and browsing using metadata.

In the process of describing information with metadata, more than one piece of information may share the same metadata. This results in the grouping of information under some sort of categories as expressed by the metadata. Actually, in an information and knowledge organization process, *classification* is an indispensable approach.

> *"A ubiquitous and timeless human cognitive activity is the ongoing effort each individual makes to construct a cohesive and predictable mental view of the world around her, seeking patterns by which to organize and make sense of it. This involves conceptually clustering things and ideas into named categories based on observable shared characteristics judged salient in a given context, with the resulting categories held together by some sort of mental framework of relationships; in short, classification. Our understanding of the world then, as well as our ability to survive in it, depends crucially on our innate ability to perceive and characterize the relationships between concepts, that is, to construct conceptually valid and robust classes of concepts and the relationships among them" (Green et al., 2002: vii-viii).*

Classification is the ordering of entities into groups or classes on the basis of their similarity (Bailey, 1994: 1). In order to create a good classification, one must distinguish the key characteristics on which to base the classification. Within an information system these characteristics can be defined as metadata, e.g., as keywords.

A good classification defines classes that are "exhaustive and mutually exclusive" (Bailey, 1994: 3). However, in practice, classification is subjective (Bowker and Star, 1999). If

[6] An index according to the Oxford English Dictionary is: "An alphabetical list, placed (usually) at the end of a book, of the names, subjects, etc. occurring in it, with indication of the places in which they occur." "A traditional index is in fact a map of the *knowledge* contained in a book; it lists the topics covered, by whatever name users might be expected to want to look them up, and includes salient (and *only* salient) references to those topics" (Pepper, 2002).

classification is attempted as a group work, reaching a consensus is usually necessary. This can be done through claims (metadata) and correspondence, where "classification is both a process and an end result" (Bailey, 1994: 2). The meaning of a metadata is necessarily rooted in a context and culture, and the interaction of people from different contexts with an information organization system is bound to be different. Therefore, forcing a standardization approach for information organization on a group of users is not desirable if the wish is to create an open system that also supports design correspondence.

2.4.2. Social tagging and communal categorization

A folksonomy is a new term that describes a categorization of documents using freely chosen (and subjective) terms (metadata), cooperatively by a group (Todras-Whitehill, 2005). The essence of the approach is that users create metadata for their own personal use, which are published, shared, and (re)used cooperatively in a community. The metadata determine the organizational structure of the documents; users browse and search using these metadata. These result in bottom-up, consensus-based classification structures.

Folksonomies as a concept is rapidly emerging. It is a social tagging mechanism, usually seen in web communities. Some of the applications are blogs, wikis, newsfeeds, social networks, and bookmarking tools. Practical applications of folksonomies are young and in progress[7]. This way of classification usually happens in flat (non-hierarchical) communities, where a classification structure is not imposed. Folksonomies are based on the principle that the community that organizes the collection of information is also its primary user. People in such a community cooperate spontaneously by sharing metadata. One could argue that folksonomies accurately reflect the community's conceptual model of the information because they are generated through social interactions over time. There is no authority to build up and control the maintenance of folksonomies. It is a cooperative effort.

Overall, there are no clearly defined relationships between the metadata that form the classification vocabulary. Similarly, since there are no relationships specified between metadata, associations between metadata cannot be made, which in turn has consequences for meaningful browsing of the information that resides in the system.

Some advantages of folksonomies are their community forming effects, their flexibility, and a lower threshold for getting involved. These qualities of folksonomies make the underlying concept of social tagging highly suitable for a system supporting reification and participation of communities of practice in the conceptual stage of architecture.

2.4.3. Information retrieval motives

From a design point of view, designers may have many different reasons for retrieving design information from an electronic environment in the conceptual phase of design,

[7] A website called del.icio.us (http://del.icio.us/) was created in late 2003 for users to organize and access their web bookmarks. del.icio.us contained the features of what would later be called a folksonomy, and this was quickly repeated in other 'social software' such as Flickr (http://www.flickr.com/) and Furl (http://www.furl.net/). Flickr is a site for users to organize and share their photographs. Furl is similar to del.icio.us.

such as gaining knowledge, and finding creative inspiration. Architects have reported looking at precedent libraries during the act of designing in conceptual design when they wish to change their mode of engagement “from output to input activity” (Murty and Purcell, 2003). They then look at precedents with the intention of searching for something specific, or in order to get excited and inspired, and thereby “invoke creative activity” (Murty and Purcell, 2003). They simply start exploring, think about what they see, get inspired, and go back to designing. Some architects have presented examples of a design solution that was invoked from an image from an unrelated context (e.g., Stouffs and Wieringa, 2006).

As an example, given a collection of precedents of a specific building type, e.g., theaters, one may be interested in a particular theater hall because of the architect that designed it. One may also want to look at various foyers in order to get an overview of different circulation schemes used in theaters. Alternatively, one may want to deduce rules of thumb about designing theater halls with good acoustics by looking at theater halls that are considered to be examples of good acoustics. One can also explore such a collection of precedents without any apparent pattern. Many more examples can be enumerated.

In summary, information retrieval actions of designers within information environments generally fit one of the following two categories. Firstly, one may want to retrieve a specific known document that resides in the repository. Secondly, one may want to retrieve all documents pertaining to a certain concept or topic, including their links to other related documents. Such an overview of relevant documents may provide the necessary information in order to establish or verify a certain design aspect. Especially in the conceptual design phase, the possibility of interpreting the entire document structure seeking information related to a concept of interest is an important requirement for such an environment. Effective overviews of the information structure, or part thereof, enable interpretations of the information space that may lead to new understandings and to the recognition of important aspects or entities.

Information retrieval is generally based on queries, however, queries can be posed in different ways. In most commercial information environments, a query is formulated using Boolean logic expressions and/or a proximity matching expression. A retrieval system then searches the whole collection of documents and returns a list of documents that match the specified query (Lin, 1997). This approach is not new, and its techniques have already been extensively reviewed (Belkin and Croft, 1987). In the context of conceptual design, this approach alone is not sufficient. A more flexible information retrieval approach that supports the additional needs of designers in this design phase is needed for a digital design information environment.

2.4.4. Associative browsing

Information retrieval based on the specification of queries is standard for information environments. However, designers in the conceptual phase of design have difficulty specifying an exact query (Restrepo, 2004). Furthermore, if designers are supported to browse the result set and to see the relationships between the items contained in this set, they can achieve a cognitive mode of browsing. In this cognitive mode of browsing, designers are able to recognize and follow connections between precedents that arise from shared concepts and ideas in precedents.

Thus, retrieval activities in an information system that is to be used in the conceptual design phase must support the (visual) exploration of the information contained within this system, thus browsing this information (Keller, 2005; Lin, 1997). According to Lin (1997), in relation to information retrieval, browsing is particularly useful when:

- "users are not familiar with the content of the collection and they need to explore the collection (Motro, 1986)
- users have less understanding of how information is organized in the system and they prefer to take a low cognitive load approach to explore the system (Marchionini, 1995)
- users have difficulties in articulating their information needs (Belkin, 2000)
- users look for information that is easier to recognize than to describe (Bates, 1986)"

All of these needs for browsing apply to designers in the conceptual stage of design.

Browsing is an explorative and interactive cognitive activity. Designers are able to make relevant connections among multiple precedents when freely browsing design ideas within precedents. While browsing associatively, users discover unanticipated new ideas and encounter creative coincidences (Oxman, 2004; Goldschmidt, 1995; Murty and Purcell, 2003). Browsing generates such potential because the user can see the underlying relationships between information items, and can follow different paths accordingly, with respect to the task at hand or the stage of the cognitive browsing process at that moment. This is called associative browsing: users browse using the underlying associative relationships between information entities. "The content acquired by browsing is, probably almost immediately, integrated in some way to begin forming a mental map" (Spence, 2001: 98) (Figure 2.5). This mental map is formed through perception, but has no meaning as such. This mental map is then subjected to the process of cognition in which a model of understanding is established in memory (Humphreys and Bruce, 1989). This transfer from perception to cognition takes place in the context of other internal models of the user, in other words, the resulting model is integrated into and is understood in the context of and in reference to the existing knowledge of the user (Figure 2.6).

In the design of an information system for architects in the conceptual phase of design, one must understand and formulate a browsing strategy. This will enable the system to fit its intended use context and increase its usability. Such strategies have perceptual and cognitive aspects. Spence (2001: 108) describes a matrix where a browsing strategy has, on one axis, a cognitive and a perceptual component, while the other axis consists of planned and opportunistic strategies (Figure 2.7). A designer of an information system needs to incorporate such behaviors into the design and implementation of the system, both at the information model design stage and at the user interaction design stage. A graphical interaction mechanism that supports browsing needs to be presented to the users of such systems.

Facilitating effective browsing in a system requires two components: the underlying information structure must be suitable and enhanced with an appropriate number of relationships between information entities; and the information display must be visual and dynamic. Some functional requirements in information systems for effective browsing are:

- Users should be able to position themselves in an area of interest of the information structure

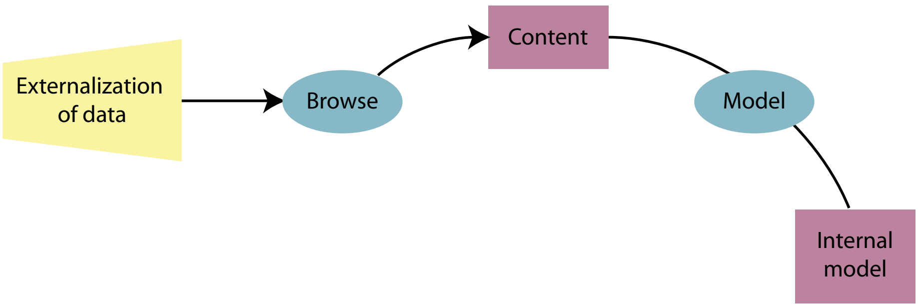


Figure 2.5. “Browsing of externalized data leads to an internal model.” Figure after Spence(2001).

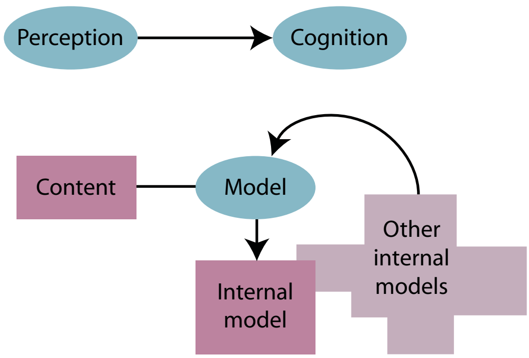


Figure 2.6. “A model relevant to a particular task may be performed by reference to an existing model and may then form part of it.” Figure after Spence (2001).

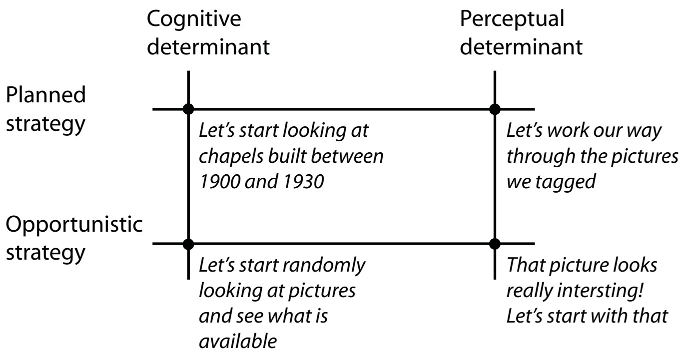


Figure 2.7. “Cognitive and perceptual determinants of planned and opportunistic strategies.” Figure after Spence (2001).

- Users should be able to recognize suitable directions in which to further browse and search
- Users should be able to move effectively and efficiently through the information structure

2.5. COMPLEX INFORMATION STRUCTURES

We aim at building digital environments for the acquisition, use, and reuse of information and knowledge where the knowledge is built up and used by members of a community of practice. As discussed in section 2.2, such environments contain collections of documents depicting precedent information, and an organizational structure that is built up by the users of the environment, giving the users the possibility to reflect on the knowledge entities and their relationships. Such structures are considered to be *complex information structures* (Tunçer et al., 2002a).

Definitions of complexity are often related to a system. According to Simon (1962), a complex system is one made up of a large number of parts that interact in a nonsimple way. A complex information structure is a result of activities performed within a complex adaptive system, especially around information organization. Information structures are created, at a minimum, by a collection of information entities, in this case, documents, an organization of these entities, and a specification of the relationships between these entities (Figure 2.8). In the context of the conceptual architectural design process, the individual documents and their mutual relationships define the information structure. The information intensity involved in the structure raises questions of complexity: how to organize and intra-relate large amounts of information in order to facilitate correspondence on this information. This involves issues of both modeling and visualizing this complexity.

The best way to handle the complexity of architectural information is not through a simplification of the information structure. On the contrary, a complex information structure that enables views unbounded by the original documents is advocated. Complexity is a necessary characteristic of information models if they are intended to yield more than a few predefined viewpoints to the information. Targeting a largely unfamiliar audience, the indeterminacy of viewpoints provides the possibility to anticipate individual requests from the audience. Unexpected viewpoints derived from the information can also invoke new interpretations of existing information, which in turn can lead to creative discoveries. An important question is how to achieve such complexity in a simple approach.

We propose the adoption of a representational language as a common syntax for describing the organizational structure, the documents, and their integration in a global information structure where there is an integrated structure of components and relationships, represented in a uniform way. A computational framework that supports this structure needs to fulfill some requirements about the representation of the documents, the recognition of structures and relationships, and the formal structure of the resulting model.

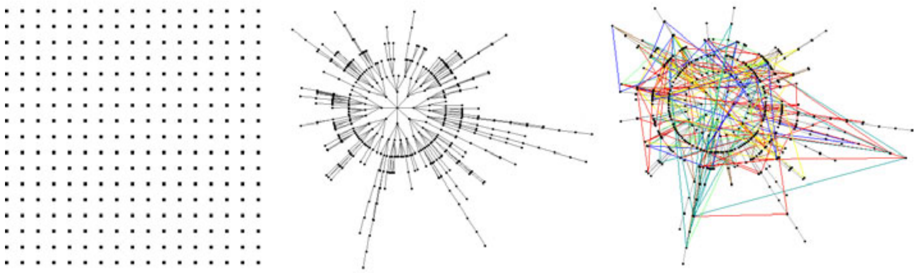


Figure 2.8. A schematic description of an information structure. Left: a collection of information entities, Middle: these information entities are organized in a hierarchical structure, Right: Further relationships are created between the information entities. Diagram after the Fake.Space project (Engeli, 2001).

The complexity of the information structure, however, should not stand in the way of its ease of use, especially when integrating individual documents into it. Therefore, the tools, mechanisms, and techniques for creating the integrated information structure should be as clear, straightforward, and intuitive to use as possible. These tools, mechanisms, and techniques must not change over time for the participants to be able to correspond on the content. The content, on the other hand, changes, reflecting on the changes and evolutions that come with the state-of-the-art of society and culture (see section 2.3).

2.6. EXEMPLARY PRECEDENT INFORMATION ORGANIZATION SYSTEMS

Precedent information and knowledge organization systems are implemented using electronic document management environments in which, commonly, a *document-based approach* is adopted. The document-based approach to information organization treats documents as information containers, containing information in some structured way, and metadata describes documents. A document is a discrete collection of data kept together by the user. A document may be composite, meaning it may contain information represented in various formats such as text, images, video, sound, etc. Documents become entities or objects that are organized and related according to different categories and attributes, categorized and hyperlinked within an organizational structure in order to support navigation through the information space. More sophisticated examples of document management environments rely on a database for storage and management of the information entities, and offer a more complex categorization of the documents and their relationships.

This approach is adopted in Electronic Document Management Systems (EDMS's), and more recently, Content Management Systems (CMS's). These systems offer functionalities related to the scanning, indexing, organizing, modifying, processing, storing, and retrieving of documents (Megill and Schantz, 1999: 81). The content of a CMS usually resides in a repository, which is accessible to all users in accordance to their access rights. This content, in the form of documents, can be of various formats, including text, raster or

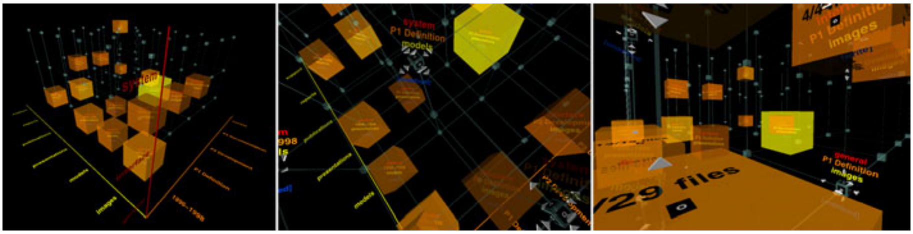


Figure 2.9. A 3-D representation of the ICCS environment.

vector based graphics, and audio or video. Documents are augmented with links, attributes, and methods for viewing this information in a variety of ways.

Recently, CMS developments are gravitating towards the web. In the form of web-based project management applications, these have also found their way into the AEC industry, providing facilities for organizing and viewing documents, and redlining drawings and images (Smith, 2000; Stouffs et al., 1998). An exemplary web-based CMS is the Information, Communication, and Collaboration System (ICCS) project (Stouffs et al., 1998) This is a CMS that is designed to support communication and information exchange within the Swiss Architecture, Engineering, and Construction (AEC) industry. Here, the documents are organized according to a three-dimensional main organization, and can be further related through relationships (Figure 2.9). Requirements for any content management system are that it is easy to manage but still offers the flexibility to tune the system to meet a broad range of user and organizational requirements.

In precedent-based systems, a general way of organizing and indexing information is through archival metadata such as name, location, style, time period, architect, etc. In the conceptual phase of design, in addition to these archival categories, a flexible system is necessary that is capable of supporting subjective information and the collective organization of metadata in order to support activities of a community of practice. A system in which users can model domain and precedent knowledge as well as organize their documents requires a significant development. In order to do this, an organizational structure is necessary that facilitates the collective handling of the definition and management of architectural concepts and their relationships.

There are a number of systems developed for computationally supporting conceptual design using collections of design documents. Some of these are merely electronic catalogues. Others are also targeted at supporting cognitive processes of designers at the early stages of design, as well as providing organized precedent libraries. Some target the cataloging, adaptation and reuse of knowledge embedded in 'cases'. These adopt the *case-based reasoning* (CBR)[8] approach (Kolodner, 1993). CBR approaches do not rely on generic

[8] CBR is a branch of Artificial Intelligence (AI) that aims at tools for assistance in the reuse of information from organized memories of past experience. It is "a formalization for the development of a computational model of problem solving that is based on memory organization and reminding. CBR has been developed as a process model with specific stages and knowledge resources that reflects the research in analogical reasoning" (Maher et al., 1995: 2). "CBR systems typically have to deal with the representation of the case content, the organization of the case memory, strategies for recalling cases, and mechanisms to modify cases to fit new problem situations" (Aygen, 1998: 23).

domain knowledge, but make use of specific knowledge of previously encountered situations: cases. These cases are stored in a case-base. CBR systems are in principle learning systems, i.e., they learn from the application of their cases in new situations. CBR systems match solutions based on similarity.

A case becomes memorable as a precedent when it makes a particular conceptual contribution to design. Within the scope of this research, the interest is on precedent based systems. Some relevant examples of environments that organize precedent knowledge and information are described and discussed below. These examples are taken from the fields of architecture and industrial design. Some other systems which have not been elaborated and discussed here are CADRE (Dave et al., 1994), ARCHIE-2 (Domeshek and Kolodner, 1992), IDIOM (Smith et al., 1995), and SEED (Flemming and Woodbury, 1995).

Alvar Aalto: a conceptual analysis: Madrazo and Weder (2001) have developed an interactive website of students' analyses of works of Alvar Aalto[9]. The organization consists of three modes: descriptive, analytical and associative. The associative mode organizes and indexes the information discovered and analyzed by the students according to a number of concepts, represented as keywords. These concepts were conceived during a brainstorming session in which all students and instructors participated. A number of common concepts were defined from the similarities of all the analyses done by the students. There were no further semantic relationships defined between the concepts, they are organized simply as a list. Users can freely navigate through the information using the concepts for browsing. This is an interesting system, but its purpose is solely to archive and present the information. It does not have possibilities to create an open and interactive system.

Electronic Design Assistance Tool (EDAT): EDAT (Akin, 2002; Akin et al., 1997) is an educational environment that presents and organizes design precedents collected by students in the early phases of a design studio project according to various characteristics. It additionally offers the students a tool to present their work in the design studio and is extendable in different ways, e.g., for carrying out performance analyses on the stored test cases. EDAT enables students to organize and index design documents in a clear and comprehensible way. EDAT allows its users, the students, to construct and organize a repository of precedents of the same building type as the project at hand, which can be consulted in the design process. "This is one of the greatest strengths of EDAT: its basic identifiers, or indexing of subject matter, are completely user defined. It does not assume that information must be organized in any particular way. However, this places on the case-builder the burden of creating a coherent topic tree. Since the manner in which facts are indexed has a great effect on how these same facts are retrieved from the case base, the conceptual process of designing a topic tree for each building type lies at the heart of both information retrieval and information storage in EDAT" (Akin, 2002: 427). "Browsing the database in EDAT is accomplished through several filters. The filter criteria are building type, building name, architect, and topic. The criteria can be applied in any order and any or all can be omitted" (428). The user interface of EDAT can be improved to be more user friendly. It does not support cognitive browsing. Its organizational structure does not allow for the definition of a semantic structure that involves semantic relationships. This would

[9] http://caad.arch.ethz.ch/aalto/

allow the definition of chunks of knowledge, perhaps as typologies, in a more powerful way.

A Dynamic Architectural Memory On-line (DYNAMO): DYNAMO (Heylighen and Neuckermans, 2000) is an educational design assistant tool. Its goals and underlying pedagogical principles are similar to EDAT. It is an extensible precedent library that offers a communication platform to students for sharing and exchanging information, design ideas, and insights. The tool is web-based[10]. Just like EDAT, DYNAMO does not possess an organizational structure that allows for the definition of a semantic structure and therefore does not support cognitive browsing.

ProductWorld: Muller and Pasman (Muller and Pasman, 1996) have created a precedent library that was organized according to a predefined typological structure organized as a taxonomy of keywords. This typological structure consists of a trilogy: typology of function, form, and meaning (Pasman, 2003: 59). It is assumed in this approach that the typological categories inherently organize images according to the knowledge they contain, because this knowledge and its relationships with other knowledge is already included in the classification structure. One problem with this approach is that it does not allow for correspondence among the users of the system, it is just an aid to get ideas and inspiration. Additionally, a mediator always needs to update the categories, the knowledge classification is not very dynamic and interactive.

PRECEDENTS: Oxman and Oxman (Oxman, 1994a; Oxman, 1994b; Oxman and Oxman, 1993) developed PRECEDENTS which is a case-based design aid for architecture. PRECEDENTS targets use in the conceptual design phase by providing a representation of the conceptual knowledge in the precedents. This knowledge is formalized by means of cognitively based "design stories" rather than the cases these describe. Design stories consist of design issues, concepts and form as the indexing scheme of the cases. The explicit mapping of these results in a network structure that can be searched and browsed. The system is filled by acquiring knowledge through reading and analyzing critical writings. This requires a huge work load to populate the system with knowledge. Unlike some other case-based reasoning systems, PRECEDENTS can be considered less 'objective' because the knowledge extracted is not 'pure', but interpreted by the author into design stories. This can be considered as an enrichment of the knowledge. However, with respect to being used in real life in a design situation, the entry and manipulation of stories and cases in the system is greatly challenged by the complexity of the indexing system.

Visual Understanding Environment: The Visual Understanding Environment (VUE) is an open source project based at Tufts University[11] (Kumar and Kahle, 2006). VUE is a general purpose concept mapping application. It is mainly an educational initiative focused on structuring, presenting, and sharing digital information. VUE has capabilities for creating attractive visual concept maps, but it does not distinguish between the organizational structure and the body of documents. Therefore, creating a semantic structure consisting of cognitive structures independently of a collection of documents is not easily achievable using VUE. However, because VUE is an open source initiative, it is an excellent candidate for being extended and adapted to implement applications of the framework developed In this research.

---

[10] http://dynamo.asro.kuleuven.be/

[11] http://vue.tufts.edu/

2.7. AN ASPIRATION FOR ARCIMAP

After getting familiarized with digital environments for precedent-based design, an educational scenario that we come across regularly in our academic activities is introduced. This scenario demonstrates the need for the definition of a framework for information and knowledge organization that allows users to explicitly record and organize the knowledge that resides in the stored design documents within digital design information environments.

At the beginning of a design project, design teams usually gather and collect relevant information in order to familiarize themselves with the project context, and to set the problem framework and define the design issues in which they will operate. This information is (implicitly or explicitly) organized around a number of aspects. The designer(s) already has notions and ideas on these design aspects even before the information collection activity. However, the collected information definitely provides cognitive support for problem setting. The cognitive framework of the design problems, entities and relationships forms partially during the information gathering and organization process. The aspects that are used to organize the collected information can demonstrate this cognitive framework if represented in a suitable manner.

We have developed such a cognitive framework as part of a collective information gathering and brainstorming activity within the context of a design exercise with students[12]. The design exercise was the redevelopment of an urban site in central Rotterdam (see Tunçer et al., 2005 for a detailed description of the project). In the first workshop (that lasted a full day), the students got familiar with the site and the related considerations in order to define a project description and program brief themselves. In the first part of the workshop, they concentrated on the information gathering activity. They analyzed and investigated the site and the surroundings. Some issues students concentrated on were urban aspects, accessibility, functionality, views and daylight issues. Students first collected information about the site in the form of plans, photographs, websites and articles. They also made notes and sketches about a preliminary concept that they had developed for the site.

At the beginning of the second part of the workshop, all students and the instructor sat around the table and ran a brainstorming session. The goal of this session was to define a collective framework describing the important aspects that play a role in the preliminary design stage for this specific site, and the relationships between these aspects. This session started by each student presenting the gathered information, describing their preliminary design concept, and a short interactive discussion about the presented concept. Then, the whole group participated in a discussion of defining and relating design and analysis aspects for problem setting. These aspects were noted on a large paper, one of the students volunteered to be the mediator and recorder of this process. Next, another round around the table was made, this time each student named a number of necessary aspects that needed to be taken into account, and these were discussed and written on the paper. At the end of the round, all relevant aspects had been collected. This collection reflected on each student's interest and viewpoint. Then, the students collectively defined the

[12] An elective M.Sc. course that was offered at the Faculty of Architecture, Delft University of Technology, in the fall semester of 2004 to eight students. The name of the course was Mediated Discourse.

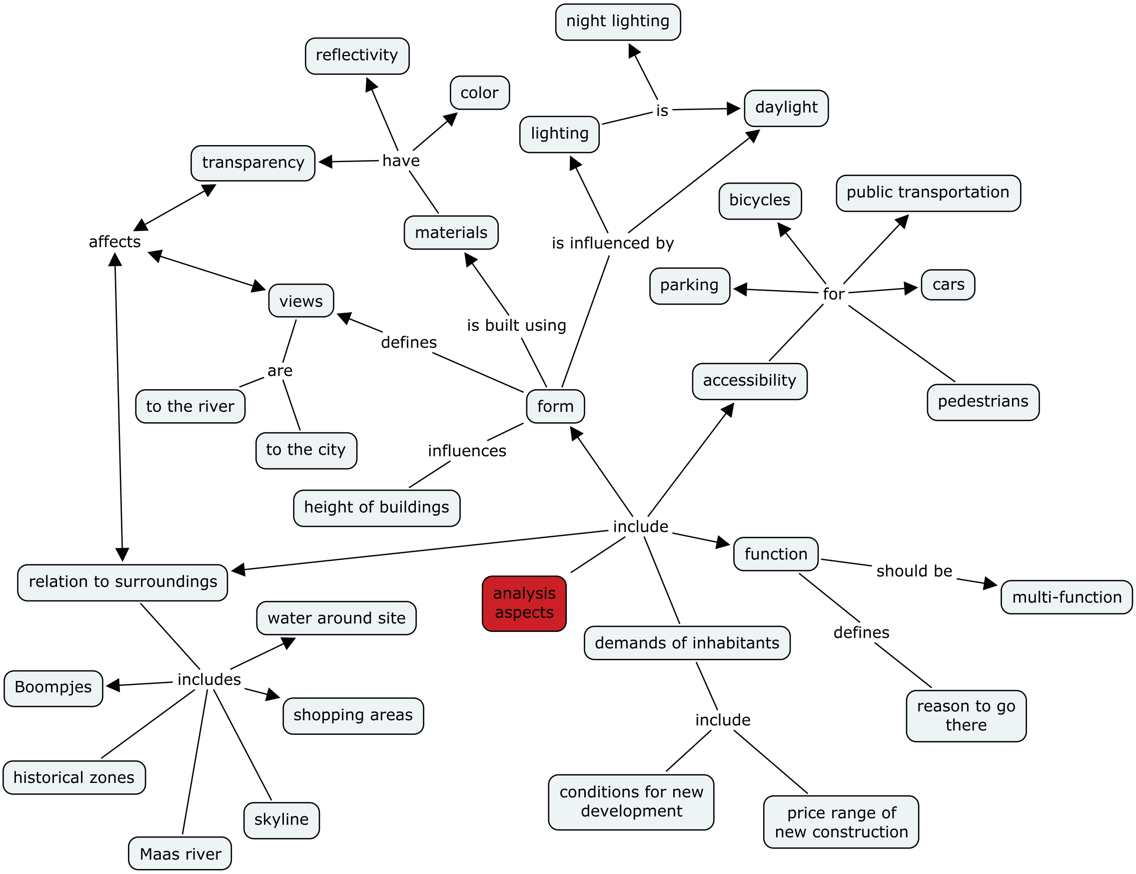


Figure 2.10. The final concept map defined by a group of third year students at the Faculty of Architecture, Delft University of Technology. This concept map was developed in a design analysis process.

relationships between the aspects. First, a tree structure emerged. Then, students started defining more associations between various branches of the tree. This defined the final visual structure of concepts and relationships defined as a concept map[13] (Figure 2.10). The students stated that some of these aspects and relationships came up from their previous knowledge as designers, and some they learned by searching for and collecting information. In any case, the resulting structure represented their collective cognitive framework for problem setting for a design for a specific site.

When this concept map acts as the organizational structure for the documents that the students collected in a digital environment, the defined aspects and relationships in this map can be assigned to each collected document in order to index and store these documents within the environment. The resulting collaborative complex information

[13] Please see section 3.2.1 on concept maps.

structure acts as a record of the cognitive process, and can be reused by other designers interested in a similar project and process.

In this research, a computational framework called **ArcIMap (Architectural Information Map)** has been developed that enables architectural communities of practice to create complex adaptive systems creating complex information structures and acting on these systems and structures, firmly rooted within their specific social and professional contexts. ArcIMap defines processes, representations and techniques to achieve this. Teams of designers record their cognitive processes in a semantic structure, and use this structure for the organization of collected information entities as documents. These activities fit within their regular interaction and work processes. All these aspects will be described, discussed and tested in the rest of this dissertation.

2.8. CONCLUSION

In this chapter, the importance of collectively and collaboratively constructing information collections and creating organizational structures for these collections has been discussed within the context of architectural communities of practice active in the conceptual design stage. Our goal is to design and create digital environments where members of such a community of practice can perform reification and actively participate and therefore correspond on design information, where the information in the environment is both the means and the result of social processes. In order to achieve this goal one needs to create a framework for information and knowledge organization that allows users to explicitly record and organize the knowledge that resides in the stored design documents within such environments. This framework must compile representational, technical and procedural components in order to achieve our goal. Since the meaning of information and knowledge is deeply rooted within its context, so is its organizational structure. Therefore, the purpose of this framework cannot be the definition of a taxonomy (or a similar structure) within a knowledge domain. The framework must provide a flexible and extensible 'recipe' to design and implement complex information structures within various design contexts and for various architectural communities of practice.

As mentioned in the previous section, ArcIMap is a computational framework for the design and implementation of digital conceptual design information and knowledge organization environments for a community of practice within a given context. ArcIMap defines representational, technical and procedural components that enable this. A digital design information environment that implements the ArcIMap framework must store documents and build up (precedent) libraries possessing a large amount and variety of information, and be extensible, flexible and easy to use. It must enable the users of the information to also be its builders by collectively constructing an organizational structure for knowledge and information with an unrestricted encoding of personal notions and insights. A representational language must serve as a common syntax for describing the organizational structure, the documents, and their integration in a global information structure where there is an integrated structure of components and relationships, represented in a uniform way.

The following chapter presents *complex information structures* as the foundation of the representational, technical and procedural components of ArcIMap. The elements of complex information structures are defined together with processes, mechanisms and technologies of their design and creation.

3

COMPLEX DESIGN INFORMATION STRUCTURES

Complex design information structures result from information processes carried out by architectural communities of practice in the conceptual design stage of architectural design[14]. Architectural communities of practice generate and handle common knowledge by reifying and recording knowledge into documents in order to support cooperation and mutual understanding of informal group activities among members, and actively participate in social processes in order to personally contextualize this recorded knowledge. Reification and participation lead to correspondence within the community and result in complex information structures. These structures are comprised of documents depicting design information and consider an organizational structure that is built up by the members of the community giving the members the possibility to reflect on the knowledge entities, using metadata to describe these documents and a specification of the relationships between these documents. Complex information structures are embedded in digital design information environments for the acquisition, use, and reuse of (precedent) knowledge where the knowledge is built up and used by members of a community of practice. When building such structures, both modeling and visualizing the complexity require careful consideration.

A complex information structure is the embedded foundation of **ArcIMap (Architectural Information Map)**. ArcIMap has been developed as a computational framework for the design and implementation of digital design information environments in the conceptual design phase for architectural communities within specific social and professional contexts. Complex information structures possess representational, technical and procedural components that make ArcIMap possible.

The adoption of a representational language as a common syntax for describing all kinds of information structures and their integration into a global structure can enable the creation of complex information structures. In order to do this, two techniques are

[14] Please see the previous chapter.

proposed: the separation of the document structure and semantics[15], and the decomposition of documents by content. These techniques increase the structure's cardinality and its interrelatedness towards a more complex structure by augmenting the structure's relatedness with content information, and by expanding the structure through the replacement of document entities by detailed component substructures.

In this chapter complex information structures are described from representational, procedural and technological points of view.

3.1. SEPARATING DOCUMENT STRUCTURE AND SEMANTICS IN COMPLEX DESIGN INFORMATION STRUCTURES

Complex information structures at the same time result from and enable digital environments for the acquisition, use, and reuse of design knowledge in the early stages of design where the knowledge is built up and used by members of a community of practice (see Section 2.5). As discussed in section 2.2, such environments contain collections of documents depicting abstractions of information, and an organizational structure that is built up by the users of the environment, giving the users the possibility to reflect on the knowledge entities and their relationships. Complex information structures are comprised of a collection of abstractions, an organization of these abstractions, metadata describing the abstractions, and a specification of the relationships between these abstractions. Complex information structures have a high intensity and density in terms of amount and relatedness of information. ArcIMap defines and enables the building up of a complex information structure.

An architectural abstraction has been defined in section 2.4 as an abstraction of the content of a design object, such as function, acoustics, structure, process, form generation, space, and organizational relationships (see Figure 2.4). An abstraction itself can be understood in a syntactic manner as a composition of components, their metadata, and relationships between these components (Mitchell, 1994) (Figure 3.1). While each abstraction touches upon a different aspect, abstractions relate through commonalities, similarities, and variations in vocabulary, that is, the components and their relationships. When the abstractions are numerous and diverse, recognizing these relationships between (components of) abstractions creates a tight network in which the individual abstractions no longer stand out. Metadata describing these components and relationships add an extra layer of relatedness of abstractions. Such a network of abstractions can be said to embody a complex information structure. A complex information structure enables a

[15] Semantics is defined in the Oxford Dictionary as: 1 the branch of linguistics and logic concerned with meaning. 2 the meaning of a word, phrase, sentence, or text. In a broad sense, in linguistics, semantics is the study of meaning of expression, as opposed to syntax, which deals with the formal structure of expression, and as opposed to pragmatics, which is the study of the influence and contribution of contextual factors to the meaning of expression (Croft and Cruse, 2004). In computer science, semantics is an abstraction based on mathematics and logic, with the purpose of analyzing and verifying programs and systems. The semantics is the "formal specification of the meaning and behavior of something" (Rumbaugh et al., 2005: 580). For example, a UML diagram of a system or program denotes its semantics. Unlike in linguistics, in computer science, meaning at a lower level is unambiguous and does not depend on the context. Then, a language's syntax defines the spelling of language components and the rules controlling how components are combined. It is said that a syntax error occurs if one misspells a command. On the other hand, if one enters a legal command at the wrong place or context, it is denoted a semantic error.

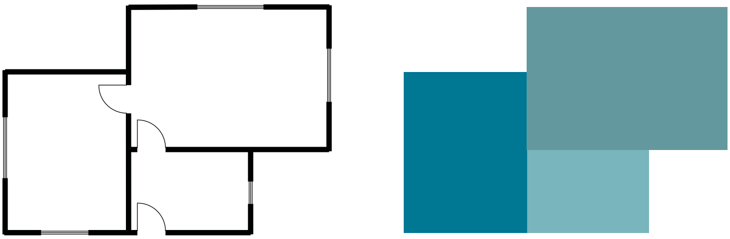


Figure 3.1. Left: A simple floor plan consisting of three rooms. Right: The vocabulary of this floor plan consists of rectangles and colors. The colored rectangles define the components, i.e., the rooms. The way these rectangles are placed on a plane defines the relationships between the components: e.g., their adjacency and access from each other.

contextual interpretation beyond the information as contained in the individual abstractions and offers powerful support for accessing the information space at both the entity level and the overview level. An implementation of such an information structure in a system would serve well in both educational and practice contexts.

The representation of an abstraction within such a complex information structure requires the definition and recognition of its components, metadata, and relationships. Components can be recognized as representations of knowledge entities (e.g., Figure 3.2). Components may be grouped, resulting in concept (component) to concept (meta-component) relationships (Figure 3.3). Considering the complex nature of design, components may also belong to more than one meta-component. Search and recognition mechanisms can assist the user in relating components within and between abstractions. Different mechanisms may be appropriate for different types of relationships. This process of relating abstractions gains meaning through descriptions using metadata.

The process of building such a complex information structure must take work processes of communities of practice into account and support reification and participation (see Section 2.3). Additionally, since the content of this structure is both a means and a result of social processes of a community of practice, i.e., a complex adaptive system, robust mechanisms must be supplied in order for the participants to be able to correspond on the content. Since the organizational structure of knowledge is at least as important as the amount of available knowledge in understanding any particular knowledge domain (Baron and Steinberg, 1987) (see Section 2.2), the collective creation and maintenance of an organizational structure must be actively enabled by these mechanisms. Additionally, flexibility is required in order to allow the content to change and reflect (on) the changes and evolutions that come with the state-of-the-art of technology, society and culture (Stouffs et al., 2004a). Extensibility and ease of use are also important requirements for a complex information structure.

Flexibility is also needed to representationally define the vocabularies that express these abstractions. Representations of concepts recognized in the abstractions must be expressible in the representational vocabulary and recognizable within the structural forms of the abstraction's representation. Since the information structures may be hierarchically organized, where a component is composed of subcomponents, the

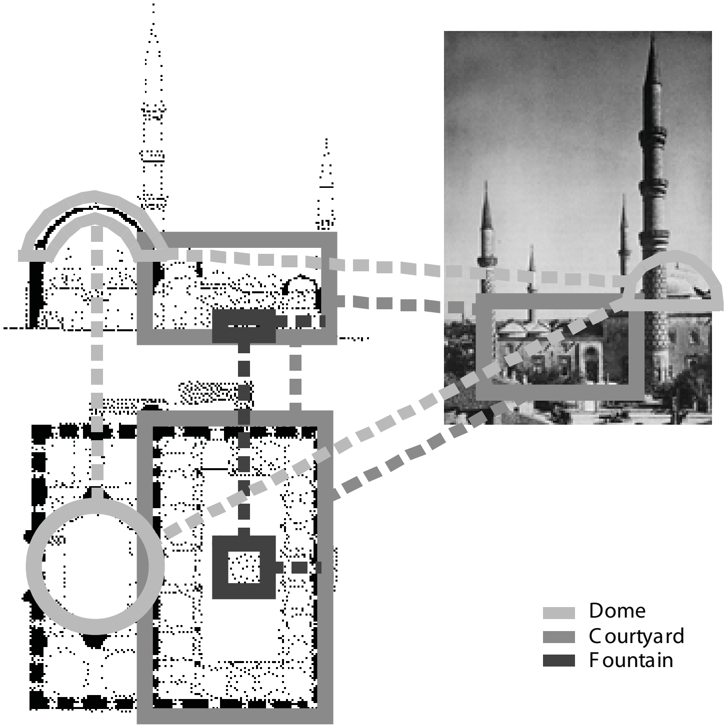


Figure 3.2. A collection of abstractions, some components and their relationships. Images from Goodwin (1971).

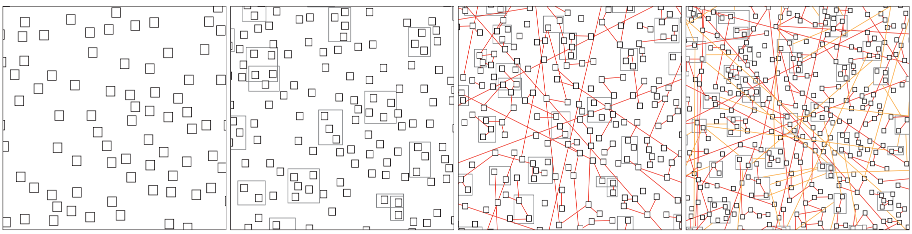


Figure 3.3. The integrated structure of a collection of abstractions. a) components, b) components grouped into meta-components, c) relationships between components, d) relationships between components and meta-components.

representation must allow for the recursive definition of components or structures. In this way, a complex representation can be achieved without imposing a fixed frame of reference.

As discussed in section 2.4.1, conceptually clustering things and ideas into classes of named concepts and the ability to perceive and characterize the relationships between these concepts is a natural human cognitive activity (Green et al., 2002: vii-viii). Within a discipline, members commonly share a definition and classification of common concepts. This structuring of shared knowledge through common concepts and their relationships gives insight into that particular discipline (Leupen et al., 1997). For example, architects generally classify building designs using types[16], based on spatial and formal features, and use this classification to communicate shared knowledge.

Such named *concepts* and their *relationships* can be considered as a "semantic structure" and this acts as a backbone for the organization of knowledge and information within a complex information structure (Figure 3.4) (see Section 3.2). 'Semantic' in this case denotes having to do with the meaning of information rather than its structure. A semantic structure constitutes an indispensable building block of a complex information structure as it is also its organizational structure. It is made up of concepts and their relationships organized as a flexible and extensible semantic network (see section 3.2.3).

A complex information structure contains, beside the semantic structure, a collection of abstractions, represented as documents. This is called the "document structure" (see Section 3.3). Documents are interpreted and broken up into components, and these components within and between documents are related, and these relationships added to the representation. Elements of the semantic structure (concepts and relationships) describe these documents or parts thereof. This is achieved in an information environment by defining elements of a semantic structure as metadata, e.g., as keywords, and assigning this metadata to abstractions or parts of abstractions (i.e., components and meta-components) (Figure 3.4).

Separating the organizational structure from the semantic structure encoded in a collection of design documents provides the flexibility that is needed to consider the evolution of the shared knowledge within the discipline. These encoded semantics, together with the explicitly defined domain knowledge, is described in a virtually separate structure from the collection of design documents. Additionally, this separation allows the semantic structure to augment the organizational structure of the documents without imposing a specific representational structure. This separation ensures extensibility and flexibility of the overall representation without imposing a fixed frame of reference, as the semantics can easily be altered without an adaptation of the documents' structure. Concepts in the semantic structure can be organized according to their relationships and dependencies, and then associated with documents and their components, or vice versa, or these can be built up in parallel. This flexibility in associating concepts to documents avoids a rigorous and tedious process of adjusting the semantics to the organization and vice versa. Users are able to build or adapt the semantic structure during the process of building the information structure.

[16] See section 5.1.1 for an account on types and typologies.

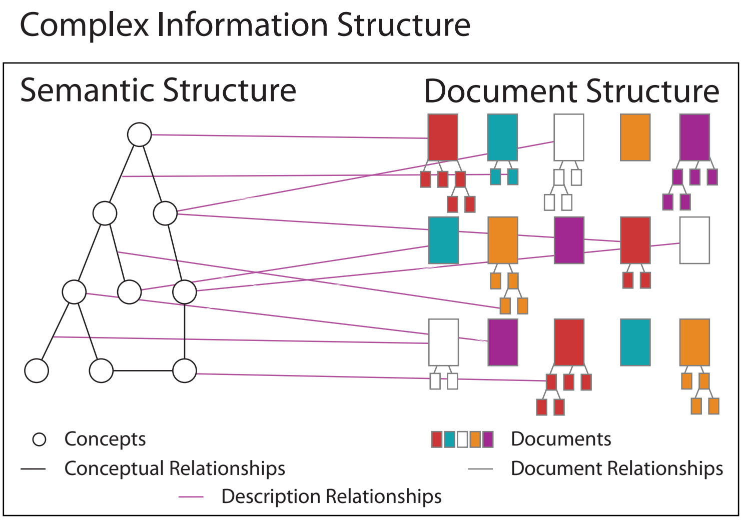


Figure 3.4. A diagram depicting the representational structure of documents and semantics in a complex information structure. The semantic structure on the left consists of concepts and conceptual relationships. The document structure on the right side consists of documents and document components related through document relationships. Concepts and conceptual relationships describe documents through description relationships. Concepts and conceptual relationships act as metadata in this structure.

The separation of the document structure and the semantic structure also offers new possibilities for accessing, viewing, and interpreting the information; most importantly, one can access the information structure from alternative views to those that are expressed by the individual documents. Specifically, it answers to a need for an information organization that enables an outsider to access this information effectively, independent of the viewpoint of the person who conceived it. First, a separation of document structure and semantic structure allows one to access specific information more directly than if a traversal of the document hierarchy is required; individual documents can be reached and retrieved more quickly when provided with more relationships. For example, in the yellow thread in Figure 3.5, the two yellow documents are related because they are described by the same concept in the semantic structure. Second, documents can be considered from a different point of view. The location of a document in the structure is no longer only defined by its place in the document hierarchy; instead, documents provide direct access to other related documents, forming a part of the first document's view. For example, in the green thread in Figure 3.5, the green document and the document components are related because they are described by concepts that are related. The degree to which this relatedness is considered is left up to the intended use of the application. Third, one can access the information structure from alternative views to those that are expressed by the individual documents. New compositions of documents and relationships offer new interpretations of the structure and generate views not inherent in the structure as created by the original documents. For example, compilations of documents and document components that cannot be easily predetermined according to given criteria can instead be selected or derived from existing and derived relationships in the structure. This can lead to new abstractions.

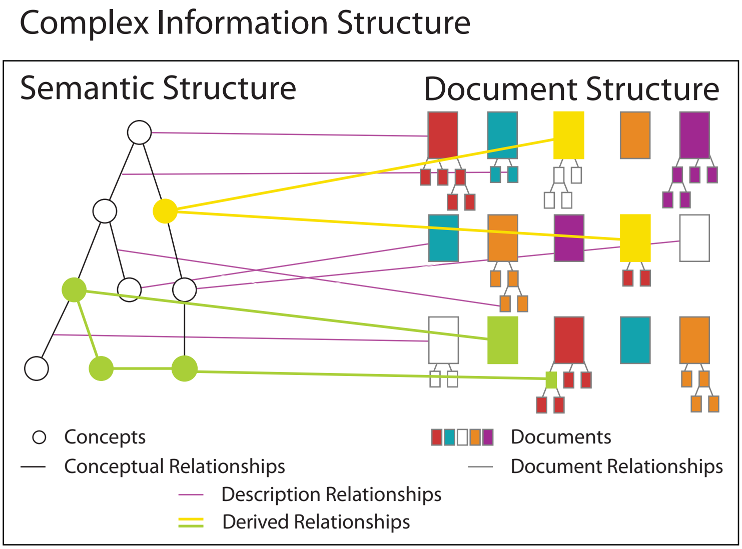


Figure 3.5. A diagram depicting the representational structure of documents and semantics in a complex information structure. The yellow thread depicts the derived relationship between the documents that represent the same concept. The green thread depicts the derived relationship between two documents that represent concepts that have a second degree relationship in the semantic structure. The level to which this relatedness is considered is left up the application.

A complex information structure offers better support for searching and browsing. Searches in a larger structure will offer more results while a denser (richer) structure assists in distinguishing entities by their relationships. Browsing a structure is also facilitated by its density as additional relationships offer more ways to move through the space. This density is a direct result of the specification of the information space by its authors.

3.2. THE SEMANTIC STRUCTURE

A semantic structure is made up of concepts and their relationships organized as a flexible and extensible semantic network (Figure 3.6). It acts as a backbone for the organization of knowledge and information within a complex information structure. Relationships that are specified between the elements of the semantic structure and the documents form the basis of the information structure. Relationships that exist between concepts automatically enrich and tighten the network of relationships within this information structure.

The concepts and relationships in a semantic structure are defined by a user or a group, belonging to an architectural community of practice working on a common project. The resulting semantic structure is project and/or community (and therefore institution) specific. It defines a common language among its users and enables and encourages reification and participation activities of a community of practice (see section 2.3). It enables and encourages reification because members of a community of practice

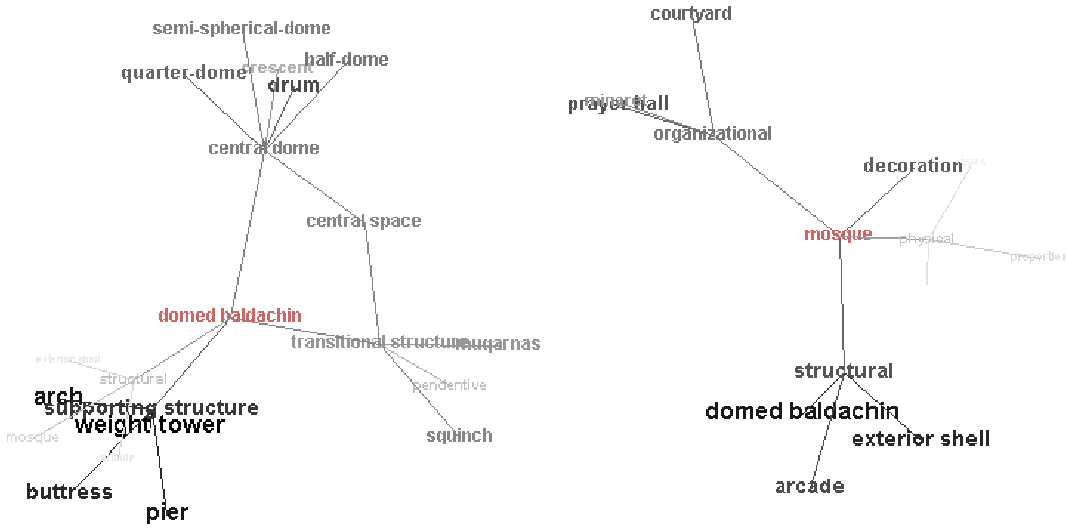


Figure 3.6. An exemplary semantic structure developed within the context of a design information environment with the purpose of organizing a collection of documents of three Ottoman mosques from the classical period (see Section 5.1). The interactive visualization has been developed using the ThinkMap application.

correspond and reach agreement on the definition of common concepts and their relationships around a topic of interest. It enables and encourages participation because the process of defining a semantic network is a largely democratic one. Active participation is possible, encouraged, and in fact needed in this process. Furthermore, the semantic structure is not a static one; it changes over time as the vision of the members change or as more information and insights become available to the group. The members of the group working on the structure also do not have to remain unchanged. The creation of a semantic structure by a group process is exemplified in section 2.7; alternatively, it can be imported from another context, such as a thesaurus.

Relationships between concepts constitute the semantic structure defined by these concepts. The form of this structure, however, is not predefined. It may be as simple as a linear structure, such as a chronological list of project phases. It may also be a hierarchical structure of concepts offering various levels of detailing. Furthermore, parts of the hierarchy may be reused as leaf nodes at various locations, resulting in a network structure, where elements can have more than one 'parent' (Figure 3.7). Concepts within such a network may be further individually related, creating an even more complex semantic structure. The structure's complexity can be extended or reduced according to particular cases. The overall structure may also constitute a combination of networks, hierarchies and linear dependencies, describing different aspects or parts of a conceptual knowledge structure. In this case, the individual structures may be considered as different dimensions within the semantic model.

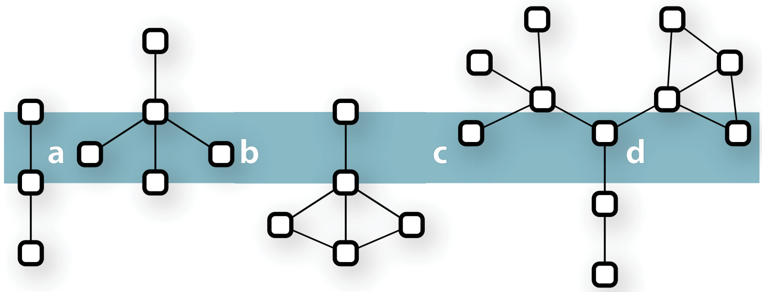


Figure 3.7. Schematic diagram of four different semantic structures for descriptive concepts. a) a linear structure, b) a hierarchical structure, c) a network structure, d) a combination of the previous structures.

The process of building up the semantic structure is crucial for a community of practice in the context of knowledge creation and sharing and social processes. Therefore in the rest of this section, methods, tools and techniques for the creation of semantic structures are explored.

3.2.1. Concept mapping for building semantic structures

Concept mapping is an excellent medium for creating semantic structures within a complex information structure. Concept maps are intuitive visual tools for organizing and representing knowledge[17] (Novak, 1998; Kremer, 1997). "Concept maps have a long history of being used in support of learners and, in general form, to support a wide variety of visual thought processes in individuals and groups"[18][19] (Gaines and Shaw, 1995). Concept

[17] A Mind Map (Buzan and Buzan, 1996) is similar to a concept map, the difference being that a mind map should have only one central concept, while a concept map can have several. Also, a mind map should be represented as a tree, without cross-links.

[18] In general, concept mapping can be used in various contexts and for various purposes, such as, to: generate ideas, for example, during brainstorming; create a clear overview of domain knowledge; communicate knowledge to others; share knowledge with others; extend knowledge by adding new knowledge easily; develop or address an understanding of a topic or knowledge; diagnose misunderstanding of a topic or knowledge; explore knowledge and internal relationships; access existing knowledge; assemble new knowledge; design structures or processes; assist problem solving by providing a good overview and relationships between items; and, fix learned material in long-term memory.

[19] In addition to defining the concepts and modeling their associations, one can utilize the added value of using colors and various shapes for concepts and links to identify differences in information. Novak (2002) specifies a process to construct a good concept map:

1. Start with a domain of knowledge that is very familiar to you (beginners should start with a limited domain of knowledge)
2. Identify a particular problem or question that you are trying to understand
3. Identify the key concepts that apply to this domain, redundancy is not important at this stage.
4. Rank and group the concepts in an order of most general or most inclusive to the most specific. This ranking can be approximate. Concepts may fall into more than one group.
5. Construct a preliminary concept map in a hierarchical structure. Use a medium where you can move concepts around easily.
6. Search for cross-links between different knowledge domains on the map
7. Revise the map, position concepts in a way that is clear

maps can be used to model concepts and relationships in a domain. They consist of nodes and links, where the nodes represent concepts, and the links represent relationships between concepts. Concept maps allow humans to see all the knowledge at a glance[20]. In a concept map, the nodes and, usually, the links are labeled. The label of a link defines the nature of the relationship between the two connected concepts. Links in concept maps can be non-, uni- or bi-directional. Concept mapping, which is the strategy to develop concepts maps, is "a technique for externalizing concepts and propositions" (Novak and Gowin, 1984: 17). Figure 2.10 shows an exemplary concept map.

Concept mapping reinforces knowledge communication and shared understanding among members of an architectural community of practice. Concept maps "provide a complementary alternative to natural language as a means of communicating knowledge" (Gaines and Shaw, 1995). Constructing a concept map can provide a way to expose, reflect on, deepen, and share one's understanding of a subject. Designers, both novice and expert, see cognitive and emotive benefits of cooperating in small groups (Johnson and Johnson, 1991). When working collaboratively in a group, communication and a shared understanding of the subject among the team members is crucial. Concept maps can help the collaborating members of a group to explore and agree on the meaning of concepts (Aroyo, 2001: 33). When constructing a concept map collaboratively, each team member will have the opportunity to express her understanding of the specific area of expertise within the subject. Team members will develop a better understanding of the task and project, and the team will become more effective and productive over time (Novak, 1998). Sometimes concept mapping will help reveal discrepancies of understanding among team members, which is also quite valuable (Fraser, 1993).

The process of concept mapping also enables individuals who construct the knowledge in the concept map to reflect on the knowledge and its relationships to other knowledge that resides in the map. One must formulate concepts and relationships explicitly, and by doing so, one must carefully consider the coherency and the consistency of one's formulation of the knowledge, reinforcing one's knowledge or one's understanding of the issue that is being made explicit. The connections that one draws between concepts in a concept map go further than just finding connections between concepts; they generate an understanding of the ideas and questions represented in the concept map. In this context, concept maps can be useful both for experts and novices. When used as part of problem definition, concept mapping can help experts to "become aware of their tacit frames" (Schön, 1983: 311), because the constructor of the concept map plays an active role in the knowledge construction process (Aroyo, 2001: 33). When constructing or modifying a concept map, one needs to think about and decide whether a new concept one intends to

---

8. Dress up the concept map with fonts, colors, etc.

For unstructured brainstorming purposes, the hierarchical structure may not be opportune. A more free structure may be used with more associative relationships.

[20] Concept maps have been widely used in many areas including education (Lambiotte et al., 1989; Novak and Gowin, 1984), management (Axelrod, 1976), artificial intelligence (Quillian, 1968; Ford et al., 1996), problem solving (Buzan and Buzan, 1996; de Bono, 1994), organizational decision making (Eden et al., 1979), social systems (Banathy, 1991), knowledge acquisition (Gaines and Shaw, 1992), linguistics (Sowa, 1984; Graesser and Clark, 1985), and for many other purposes. In architecture, the explicit use of concept maps has been done by Oxman (2004) for the project "ThinkMaps".

add is already included within another concept, or whether it is important enough to exist on its own, and what its relationships to other existing concepts are.

Concept maps are based on the assimilation theory of learning of David Ausubel. The primary idea in Ausubel's cognitive psychology is that learning takes place by the assimilation of new concepts and propositions into existing concept propositional frameworks held by the learner. Therefore, "the most important single factor influencing learning is what the learner already knows" (Ausubel, 1968; quoted in Novak and Gowin, 1984: 40). Ausubel's theory defines meaningful learning: in order to learn meaningfully, individuals relate new knowledge to relevant concepts and relationships they already know (Novak and Gowin, 1984: 7). Meaningful learning "entails the search for underlying meanings and connections between information entities that are being learned, and the learner is personally involved in the task of learning" (Novak, 2002). Concept mapping facilitates meaningful learning. Concept maps are also considered valuable in the context of fixing knowledge in long term memory (Novak, 2002). Research evidence in the field of human memory systems supports this statement: to structure large bodies of knowledge requires an orderly sequence of iterations between working memory and long-term memory as new knowledge is being received (Anderson, 1992). Because concept mapping is used for organizing knowledge by building up concepts and relationships gradually, it helps to retain the knowledge in long term memory.

Meaningful learning and fixing knowledge in long term memory is highly important for novice designers. The technique of concept mapping supports novice designers in retaining the knowledge structures and schemata extracted from precedents and in building up the necessary experience (see Section 2.2). Additionally, because concept maps can be considered as a free and dynamic template for knowledge organization, novice designers can represent and structure knowledge in a way that is familiar and useable to themselves, whether individually or in a (small) team. A concept map is extensible, flexible, and dynamic; it is not meant to represent a static body of knowledge and can be extended and modified collaboratively over time.

3.2.2. Visualizing semantic structures

Concept maps and semantic networks are graphical displays by nature, and can be used to browse and search the information or knowledge structures directly, e.g., with the use of hyperlinks. The advantages of browsing an information space associatively have been discussed in section 2.4.4, and using graphic tools for organizing and representing knowledge has been discussed in section 3.2.1 in the context of concept maps and browsing. The purpose of visual displays is to give the user a good overview with the possibility to go deeper into the structure and improve one's understanding of the structure. Stouffs (2001) claims that information visualizations must have the following qualities:

> *"… eloquent, so as to be easily understandable; forceful, so as to be clear and outspoken; graceful or elegant; vivid, so as not to be boring; and persuasive in the argument it presents. At the same time, such visualizations must be varied and flexible. A variety of visualizations enables different viewpoints on the same or different information, potentially presenting alternative arguments or reinforcing the same*

argument. These must be flexible enough to respond to the requirements and preferences of the individual participant, enable a focus on individual issues, or present arguments that may be unexpected or otherwise difficult to grasp. Together, these visualizations should enable a more effective and efficient collaboration among the participants through a visual analysis of the information structure(s) and the underlying collaborative processes. In particular, these can serve to guide the user to zones or nodes of interest, highlight problems or issues that need consideration, determine activity centres, or illustrate complex processes."

Visualizing the semantic structure facilitates an effective use of this structure in the process of constructing and manipulating this structure. Effective visualizations that facilitate visual exploration and manipulation support the process of creating new concepts and relationships, and relating appropriate concepts and relationships to documents. Even without any control mechanism to ensure the consistency of the positioning of new concepts in the network, the clarity of the structure enables the user to better determine which location may be appropriate for placing a new concept in the network (Tunçer et al., 2002b). These visualizations may be 2-dimensional or 3-dimensional, depending on which best fits the particular purpose (Lin, 1997; Stouffs, 2001). A disc view in which the user can navigate, zoom, and pan seems to be very appropriate in the visualization of hierarchical structures (Papanokolaou, 2001). In relation to semantic structures, network displays to visualize the knowledge structure are worth concentrating on because of their graph representational model. Network displays usually have zoom and filter functionalities, and show details when requested in order to avoid clutter in the display. A dynamic visualization for viewing relationships in a network is very appropriate[21] (e.g., Plumb Design, 1998) (Figure 3.8, Figure 3.9, Figure 3.10). Such visualizations help users to learn (and memorize) the contents of the map (Lin, 1995). This allows users to locate information faster and more easily, especially over time. Such visualizations also allow users to quickly identify a starting point, because they give the users a sense of spatial location.

Additionally, concepts within the semantic structure can be visualized and depicted in various formats. When concepts are represented graphically (e.g., photos or sketches) and textually (e.g., keywords or phrases), one can browse or search an information system using any of the available representations of concepts. Since designers think visually, such flexible representations are especially interesting for browsing information, when users do not have any specific query in mind (Gross, 1995). In conceptual design, such uses are plentiful, as users are not only interested in individual design documents but in an interpretation of the entire structure seeking information related to a concept of interest (see Section 2.4.3).

[21] There are a number of systems to construct visualizations of network structures. Some of these are TouchGraph (www.touchgraph.com), Thinkmap (www.thinkmap.com), Prefuse (http://prefuse.org/), AquaBrowser (www.aquabrowser.nl), The Brain (http://thebrain.com/), and Kartoo (http://www.kartoo.com/). http://www.visualcomplexity.com/ groups a large number of applications that visualize network structures.

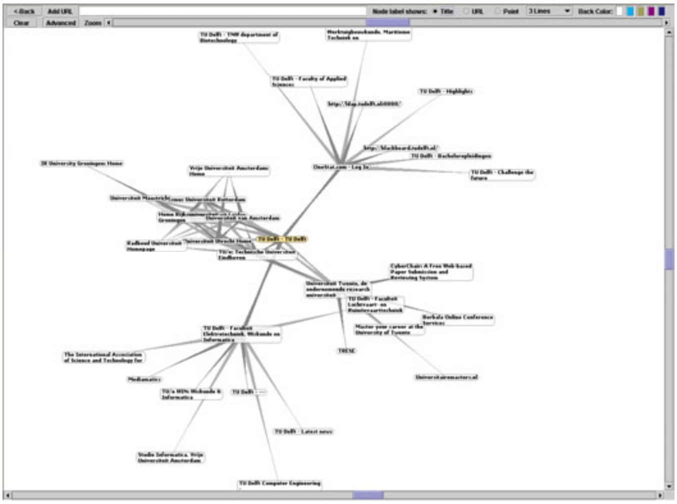


Figure 3.8. TouchGraph GoogleBrowser, with the website of TU Delft at the center.

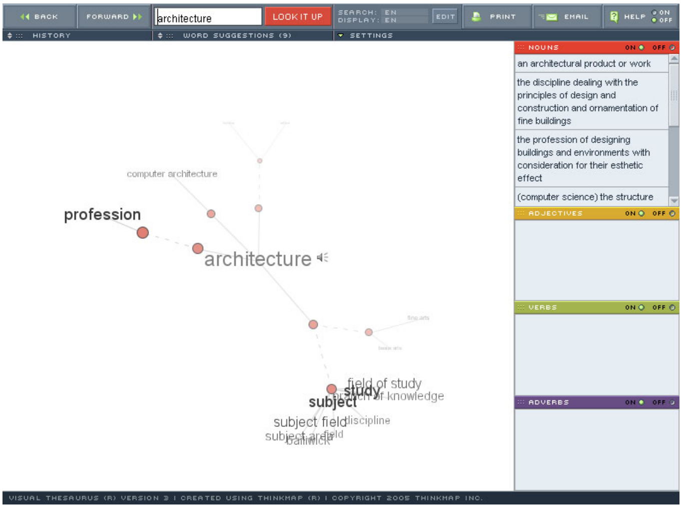


Figure 3.9. An application of Thinkmap, the "Visual Thesaurus" (http://www.visualthesaurus.com/).

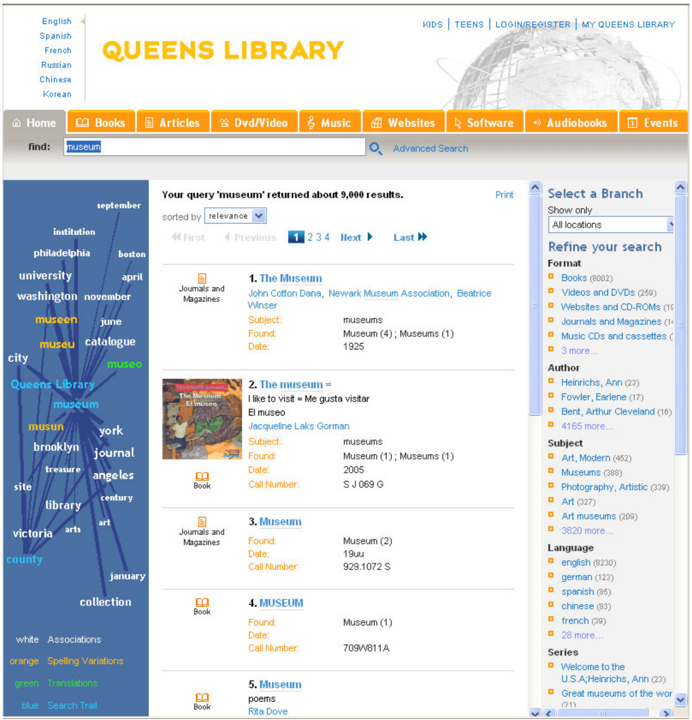


Figure 3.10. An implementation of aquabrowser for the Queens public library catalog, as a searching and browsing environment.

3.2.3. Semantic networks for knowledge representation

Semantic networks are highly suited for the computational representation of semantic structures. Semantic networks are semantic models consisting of concepts and relationships and are used for knowledge representation. Because of their graphical representation they are human and machine readable. This also enables their use as mnemonic[22] tools. Semantic networks have been incepted by Quillian (1968) as a branch of Artificial Intelligence (AI). They have been built upon concept mapping[23].

Semantic networks vary from very informal to very formal. According to Sowa (2006), "What is common to all semantic networks is a declarative graphic representation that can be used either to represent knowledge or to support automated systems for reasoning about knowledge. Some versions are highly informal, but other versions are formally defined systems of logic." Informal semantic networks are equivalent to concept maps, and

[22] A mnemonic is a memory aid. In this context this refers to the position of entities and their relative arrangement on the display acting as a visual memory aid to the users.

[23] Concept maps have been used in many formal knowledge representation formalisms besides semantic networks, such as conceptual graphs, bond graphs, petri nets, and category graphs (Gaines and Shaw, 1995).

tend to be easier to use and understand by humans. They seem to be more efficient for humans than other forms of knowledge representation such as text (Lambiotte et al., 1989). They may or may not have constraints for concept and relationship types, and even if they do, these do not need to be enforced. In such systems, ease of use is the first goal, more so than rigor. However, informal representations generally do not provide the support necessary for inference mechanisms and reasoning that may be required in knowledge-based applications. Formal semantic networks utilize knowledge interpretation and inference mechanisms and systems (Kremer, 1997) and generally enforce typed concepts and relationships. Formal representations may be quite powerful for querying the knowledge in the network, but their use by humans is not easy, and they may be quite restrictive in terms of which semantic objects and relationships are permitted in the representation. This may be a limiting factor. A successful and desired knowledge representation formalism (or tool) is one that spans informality and formality. Informality is desired because it allows the knowledge representation application to be used easily and smoothly. Some formality is needed because the knowledge stored in the system should be able to be organized and reused in a way that is not explicitly predefined.

Formally, a semantic network is a directed graph that consists of vertices that represent concepts and edges that represent semantic relations between concepts[24]. Both are labeled. Although different terminology and notations are used in different semantic network representation formalisms, the notions common to most versions are (Sowa, 1983):

- Concepts: Nodes of the graphs that represent concepts
- Relationships: Labeled arcs of the graphs represent relationships that hold between the concepts they link
- Type hierarchy: Concept and relationship types that are ordered according to levels of generality, where the hierarchy may be a tree, a lattice, or a general acyclic graph
- Inheritance: Properties of a type are inherited by all its subtypes
- Instances: Different nodes of the same concept type refer to different instances of that type

In the domain of knowledge representation, a *type* is "a specification for a set or collection of entities that exist or may exist in some domain of discourse" (Sowa, 2000: 98). For example, let's claim 'library' is a type. The definition of library as a type does not change, even though its instances may continually change: libraries in the considered context are built and demolished over time. Therefore, the definition of a type is independent of any

[24] Semantic networks and ontologies are very similar and are sometimes used interchangeably. In the context of this research, an ontology is considered as "a formal, explicit specification of a shared conceptualization" (Gruber, 1993). The semantic network technology is highly suitable to define ontologies.
An ontology in the context of knowledge management is defined as composed of a domain-specific (expandable) controlled vocabulary, a set of semantic relationships between the terms of the vocabulary, and a set of operators that control how this vocabulary represents the domain objects (Chu and Cesnik, 2001). The terms in this vocabulary have various properties assigned. The operators are defined as part of a grammar in an ontology representation language. The rules of the grammar define how to combine the knowledge in the ontology in a meaningful way. An ontology defines a meta-model, with tools and mechanisms in order to create a model within a domain. The creation of an ontology is usually based on consensus. Depending on how 'formal' the ontology is, its grammar may have varying degrees of rigor and imposed rule and structure. This degree of 'formalness' affects the ease of creation, use and maintenance, similar to any formal knowledge representation formalism. In the building industry, ontologies are used extensively for ongoing standardization efforts in the form of "product data technology" (Eastman, 1999 ; Tolman, 1999). An example is the AIA Industry Foundation Classes (IFC) initiative (http://www.ifcwiki.org/index.php/Main_Page).

changes in its instances. In a semantic network, concept types are usually organized in a hierarchy according to levels of generality: this hierarchy is called a *type hierarchy*. For example, 'building', 'public building', 'library'. Properties that hold for all concepts of a given type are *inherited* through the hierarchy by all subtypes and their concepts (Sowa, 1991: 1). For example, since all buildings need to be constructed, the property of 'needing to be constructed' is inherited by libraries. Membership to one type automatically means membership to all supertypes. Properties that hold for all concepts of a given type are *inherited* through the hierarchy by all subtypes and their concepts (Sowa, 1991: 1). For example, since all fruits need light to ripen, the property of 'needing light to ripen' is inherited by bananas. Type hierarchies are an essential component of knowledge representation frameworks, because they allow knowledge-based systems to perform inference and reasoning operations on the knowledge that is represented in the system.

A semantic network is similar to a taxonomy[25][26] (Sowa, 1991: 4; Cruse, 1986), a thesaurus[27] (Fellbaum, 1998; Miller, 1995; Miller et al., 1990), or a faceted classification[28] (Ranganathan, 1962; Tzitzikas et al., 2004), but with a formalism that enables computers to process its content (Lee, 2005). It consists of a set of concepts, axioms, and relationships, and represents an area of knowledge. Unlike a taxonomy, a semantic network allows modeling arbitrary relationships among concepts, representing logical properties and semantics of the relationships, and logically reasoning and querying about the relationships. Semantic networks can bridge the gap between procedural (or object-oriented) languages and purely declarative database systems, because similar to object-oriented languages, they have type hierarchies with inheritance, and like database systems, they are purely declarative (Sowa, 1991: ix).

In the domain of architectural design, semantic networks have been used for architectural knowledge representation. Two prominent examples are by Oxman and Oxman (1993) and Carrara et al. (1992). Oxman and Oxman use semantic networks as a framework for representing design stories in a library of design precedents, in which design issue, concept and form are linked. The lexicon of the semantic network acts as a memory index.

[25] Before describing taxonomies, thesauri and faceted classification, one needs to mention *controlled vocabulary*(Pidcock, 2003; Garshol, 2004). A controlled vocabulary is a closed list of terms. The simplest form of a controlled vocabulary is simply a random list without any semantic relationships. Terms in a controlled vocabulary should have unambiguous and non-overlapping definitions. If multiple terms are used to mean the same thing, one of the terms is identified as the preferred term in the controlled vocabulary and the other terms are listed as synonyms or aliases.

[26] A taxonomy is the ordering of a controlled vocabulary into a hierarchy. Taxonomies are predominantly hierarchical, generally using inclusion relations. Scientific taxonomies (Linnaeus, 1964 [1735])claim to be objective and universal, as opposed to folk taxonomies, which are embedded in social relations and culture (Cruse, 1986: 145).

[27] Thesauri are controlled vocabulary elements organized into a network structure. They extend taxonomies in order to be able to make assertions about subjects other than inclusion relationships. A thesaurus uses all kinds of semantic relationships, including associative relationships. Hence, a thesaurus has a lot more expressive power than a taxonomy. WordNet is an example of lexical thesaurus developed at Princeton University (http://wordnet.princeton.edu/ and http://www.cogsci.princeton.edu/~wn/).

[28] In a faceted classification, terms in a domain are divided into several aspects, or facets. Each of these facets contains hierarchically organized terms. Therefore, a faceted classification consists of a set of taxonomies. Each of the facets can be thought of as a different axis. A term belongs to only one facet. Faceted classification allows for multiple classifications of an object: the object is assigned a term from each facet. Thus, an object is described through various terms, each belonging to a different aspect of the domain. In this way, multiple navigational paths exist to an information item. Faceted classification is conceptually clearer, compacter and more scalable in comparison to a single taxonomy (Tzitzikas et al., 2004). Faceted classification is widely used in library sciences, but it is also used in software engineering for supporting software reuse (Prieto-Díaz, 1991; Henninger, 1997).

Carrara et al. represent descriptive design knowledge using semantic networks, where the nodes are replaced by frames[29].

A number of additional semantic modeling techniques and technologies that are based on semantic networks are worth noting here because some of their characteristics have been used in building the semantic structure of ArcIMap. These are *conceptual graphs* (Sowa, 1984) and *topic maps* (Pepper and Moore, 2001; Park and Hunting, 2003).

Conceptual graphs

Conceptual graphs are a logic-based knowledge representation formalism based on linguistics, psychology, and philosophy. Developed by John Sowa (1984), they evolved as a "semantic representation for natural language" (Sowa, 1983), and are "extensions of Charles Pierce's existential graphs with features adopted from linguistics and AI" (Sowa, 2000: 476). They have a direct mapping to and from natural language. Conceptual graphs implement first order logic in a graphic representation[30], which makes it more human readable. They are closely related to semantic networks.

Formally, a conceptual graph is a finite, connected, bipartite, labeled graph that consists of two kinds of nodes: besides the *concepts*, *conceptual relations* are also represented as nodes. This allows a single conceptual relationship to relate more than two concepts: arcs link the conceptual relationship to each of the relating concepts[31]. Conceptual graphs utilize type inheritance hierarchies[32]. In its original description, type hierarchies of conceptual graphs are static and given in its entirety in advance (Way, 1991: 111). However, there are examples of dynamic type hierarchies used with conceptual graphs. Concepts have a *type* and a *referent*. The referent is an instance of the type. For example, 'City' is the type and 'New York' is the referent. A concept can exist without having a referent specified, in that case, these are *generic* concepts (Way, 1991: 109). Conceptual relations have a *relationship type* and a *relation hierarchy*.

Figure 3.11 shows an example of a conceptual graph for a house. Concepts are labeled boxes, conceptual relations are labeled ellipses. The house concept has 'building' as its type, and its referent is 'house'. The house has four rooms with the type 'room', and the referents of the rooms are 'bedroom', 'living-room', 'dining-room', and 'kitchen'. The bedroom has an area specified as an interval; it should be between 13 and 18m^2. The bedroom has a color. The dining room and the kitchen are adjacent to the living room.

There are a number of conceptual graph inference engines that exploit subsumption (Corbett, 2003). Additionally, there are a number of tools to represent conceptual graphs

[29] Frames have been introduced by Minsky (1975) and are record-like structures that include their own attributes in the form of name-value pairs. These are called 'slots'.

[30] Conceptual graphs also translate to formulas in predicate calculus (Sowa, 1984)..

[31] Exactly one of these arcs must point from the conceptual relationship towards the concept. All others must point from the concept to the conceptual relationship. Every arc must link a conceptual relation to a concept: it is said to *belong* to the relation and to *be attached* to the concept. Not every concept however needs to be linked to a conceptual relation.

[32] These are represented as a lattice in conceptual graphs.

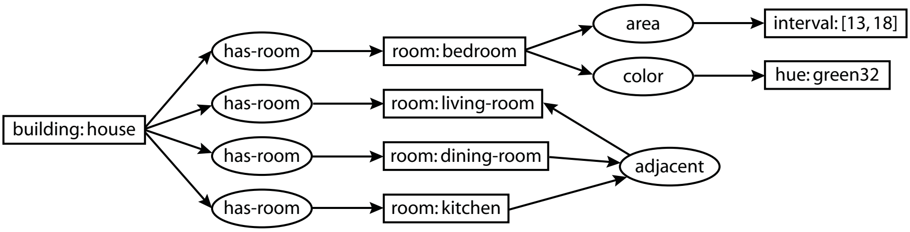


Figure 3.11. A conceptual graph example.

using XML[33] (eXtensible Markup Language) (Martin and Eklund, 1999). In architecture, the SEED project experimented with the use of conceptual graphs (Corbett and Burrow, 1996; Corbett, 2003).

Topic maps

Topic maps are an open-source semantic web technology based on XML for the representation and interchange of knowledge (Biezunski et al., 2002). The main purpose of topic maps is to bridge the gap between knowledge representation and information management (Pepper, 2002). Topic maps are founded on conceptual graphs and semantic networks, but are less formal and less rigorous. There are a number of specifications of topic maps for use with XML and the web (Pepper and Moore, 2001; Park and Hunting, 2003; Grand and Soto, 2003; Moore et al., 2005), as well as number of topic maps applications and methods to visualize topic maps (Grand and Soto, 2003), and a number of topic maps engines (Freese et al., 2003). Applications of topic maps in architecture have not been found. Dichev et al. (2003) apply topic maps in e-learning.

A topic map uses *topics*, *occurrences* and *associations* (Figure 3.12), and these can all be typed. These types are known also as the ontology of the topic map. Topics, associations, and occurrences, and their types must be defined by the creator of the topic map.

Topics: A topic represents any concept. A topic can have zero, one or more names. No enumeration of allowed names is required, instead an open vocabulary can be used. If multiple names refer to the same concept, these can be considered as synonyms. This makes every topic a synonym ring (Garshol, 2004). The fact that a topic can have more than one name offers the flexibility to use different names for a topic in various *scopes*. Each name can be given a scope, which clarifies the context. A name that has no scope is a

[33] Issues of meaningful communication and information exchange among applications on the web have brought up the need for a meta-language for the definition of device-independent, system-independent markup languages for specific purposes: XML (eXtensible Markup Language) (http://www.w3.org/TR/REC-xml: Oct 2000). Whereas HTML is used for formatting and displaying data, XML represents the contextual meaning of the data, where content and presentation are separate. By specifying a grammatical structure of markup tags and their composition (in a DTD (Document Type Definition) or an XML Schema), a markup language is defined that can be shared with other users active in the same discipline. An XML document specifies a tree of objects; applications can extract, manipulate, convert, and exchange these objects. The XML structure ensures that the data is consistently organized and is both machine- and human-readable. XML documents can most easily be presented through XSL and XSLT. XSL involves the use of style sheets for formatting, and XSLT for creating transformations of the data.

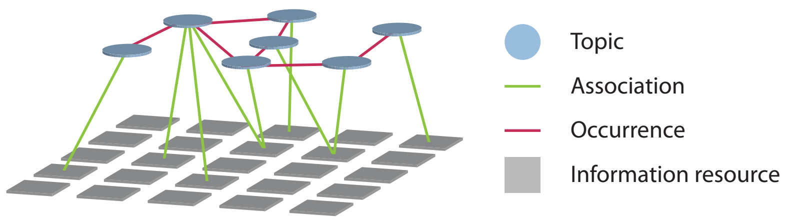


Figure 3.12. Basic structure of topic maps: topics, associations, and occurrences.

universal name, i.e., it is the same in every scope. The scope feature allows different organizations or different sections within an organization to express concepts according to their own corporate culture (Garshol, 2004). For example, if two terms in an architectural office such as “courtyard house” and “patio house” are used interchangeably, although they are used in different contexts, the scope feature allows the representation of these terms within these contexts. This supports personalization as well, because individuals can state which terminology for topics they wish to see. It also supports the grouping of multilingual terms from different languages for the same concept. Multiple topics can also be given the same name (Garshol, 2004). This presents a major distinction of topic maps from taxonomies or thesauri. However, a *topic naming constraint* states that no two topics can have the same name in the same scope.

Topics can also have types, in fact, a topic can be an instance of more than one topic types. There is a typical *class-instance* relationship between a topic and its type (Pepper, 2002). Topic types are themselves represented as topics in topic maps. This feature is missing from traditional classification techniques such as taxonomies, and assists in reasoning with the knowledge in a topic map. It helps to distinguish the kinds of terms. Type assignments to topics in topic maps are done using relationships, i.e., associations.

Associations: An association represents a relationship between two or more topics. Just like topics, they can be typed, and these association types themselves are represented as topics. Any kind of relationship can be expressed in topic maps. These relationships are generally carefully defined through the association types; they are not generic. Association types offer semantic power to topic maps. Since all kinds of relationships can be used, not only hierarchical ones, topic maps represent a network structure.

Associations in topic maps are non-directional. This is different from other knowledge representation techniques such as semantic networks and conceptual graphs. Since there is no directionality, in order to make the relationship clearer, *association roles* in topic maps describe the role of each topic in the association. Association roles are represented as topics themselves.

Occurrences: Occurrences represent relationships between topics and information resources. These information resources can be documents of any format that depict the concept represented by the topic in some way. Such occurrences are usually represented using Uniform Resource Identifiers (URI’s) where the information resource is pointed at by the URI. The use of URI’s allows the internal structure of the documents remain untouched (Pepper, 2002). Using occurrences, concepts and documents are separately represented;

they can exist independent of each other. Occurrences may also be just strings, acting as properties of a topic, in some cases providing extra information such as a phone number or an email address.

Occurrences, just like topics and associations, are distinguished by kind. These are called *occurrence roles*. They allow the differentiation of various kinds of relationships to documents. Occurrence roles are represented as simple strings, as an attribute to the occurrence. They are merely mnemonic. Occurrence roles are distinguished from *occurrence role types*; a subtle but important distinction (Pepper, 2002). Occurrence role types, just like topic types, are represented as topics, and the creator of the topic map is free to define them. Occurrence role types "characterize the nature of the occurrence's relevance to its subject" (Pepper, 2002). Occurrences can also have a scope assigned, "for example to distinguish material suitable for novices from that suitable for intermediate learners" (Garshol, 2004).

3.2.4. Semantic relationships

Concepts in semantic networks are related by semantic relationships. When members of a community of practice build a semantic structure recording their knowledge into this structure, the nature of the relationships among the concepts that are described gains great importance. Specifying the associations between knowledge entities is done by explicitly naming the relationships. There are three primary, widely recognized, meta-classes of semantic relationships (Green et al., 2002: viii): relationships of equivalence, relationships of hierarchy, and, relationships of association. All of these categories of semantic relationships are included in ArcIMap. It is important to go somewhat into detail on these categories here because these categories can act as a guide to members of a community of practice when collectively building up a semantic structure.

Relationships of equivalence

Relationships of equivalence[34] organize concepts into equivalence classes. The equivalence between the terms can be full or partial.

Synonymy: Synonyms are words with identical or similar meanings. An example of a pair of synonyms is 'elegant' and 'graceful'. The meaning of words that are considered synonyms are not always identical, but they are sufficiently similar for practical purposes. Context plays an important role in the distinction of synonyms. There are, if at all, very few absolute synonyms in a language, but they are considered synonyms in given contexts. Words in different languages can also be related through this relationship.

Antonymy: Although antonymy is not a relationship of equivalence, and furthermore, it does not precisely fit into the classification of semantic relationships presented in this section, it has been included here because of its similarities with synonymy, and because of

[34] Equivalence relationships are reflexive, symmetrical and transitive. They are reflexive, because a concept is inherently equivalent to itself. They are symmetrical because these relationships are directionally not distinguishable (Murphy, 2003: 10; Cruse, 1986), for example, the equivalence relationship between 'façade' and 'elevation' is the same as between 'elevation' and 'façade'. They are transitive, because all concepts related through an equivalence relationship necessarily belong to the same equivalence class. Furthermore, these relationships are non-hierarchical, because they do not organize concepts into a hierarchy of any kind.

its potential importance in the context of building up a semantic structure using semantic relationships. Antonyms[35] are words with opposite or almost opposite meanings. For example, 'synonym' and 'antonym' are antonyms. As with synonyms, antonyms are context dependent.

Relationships of hierarchy

Relationships of hierarchy[36] are partial ordering relationships[37]. Our most common conceptual structures are hierarchical in nature (Green et al., 2002: viii). According to Cruse (1986: 113), "the minimum requirement for a hierarchy is a set of interrelated elements structured by a suitable relation acting as a relation of dominance." In taxonomies, generalization-specification and, in thesauri, broadness-narrowness are widely applied relationships of hierarchy.

Hyponomy – Hypernymy: Hyponomy[38] is a relation of inclusion (or subsumption). It is also generally denoted as an is-a relationship. 'Concrete' and 'wood' are hyponomic to 'material'. Hypernymy is the opposite of hyponymy, for example, 'material' is hypernymic to 'wood'. A taxonomy also uses a variation of hyponomy; the elements in a taxonomy can be related by an is-a-kind-of relationship (Cruse, 2002), for example, 'oak' is hyponymic to 'wood'. Hyponomic relationships usually ensure the inheritance of properties from super-concepts to sub-concepts, also sometimes referred to as parent-child relationships.

Meronymy – Holonymy: Meronymic relationships relate concepts in a part/whole relationship. Hence a meronym denotes a component or a member of something. Meronymy is generally denoted as an is-part-of or has-a relationship. For example, 'roof' is a meronym of 'building', or 'room' is a meronym of 'house'. Holonymy is the opposite of meronymy, for example, 'Greek temple' is holonymic to 'column' whereas 'shaft' is meronymic to 'column'. Meronymy is one of the major relationships in a taxonomy.

Relationships of association

Associative relationships[39] relate concepts through an association. Although it has been attempted in the past, there is no widespread consensus about an inventory of the kinds of associative relationships (Green et al., 2002: viii). The process of defining associative relationships and their meaning can be highly personal.

Associative relationships are at least as interesting and important for conceptual architectural design as the previous two types of relationships, because architects use associations derived from analogy in their early design regularly (see Section 2.1).

[35] There are several types of antonyms: *gradable antonyms* express a contrary state, such as 'big' and 'little'; *complementary antonyms* express an either/or state relationship, such as 'dead' or 'alive' (Murphy, 2003: 46-47).

[36] Relationships of hierarchy are transitive (with certain considerations (Pribbenow, 2002: 40; Cruse, 1986: 114)), irreflexive and asymmetrical.

[37] "The term hierarchy is often used indiscriminately for any partial ordering" (Sowa, 1984: 382). A hierarchy can be a tree or a lattice; in general, any acyclic graph could be called a hierarchy. Nevertheless, a hierarchy is most commonly understood as a tree.

[38] Hyponyms are mostly applied to nouns (Cruse, 2002).

[39] Since associative relationships and their meaning can be highly personal, properties of an associative relationship (transitivity, symmetry, etc.) are dependent on the specific relationship. Associative relationships are primarily non-hierarchical.

Associative relationships are the main instruments to express such thought processes (Croft and Cruse, 2004).

Metonymy: A well known associative relationship is metonymy. Metonymy is the substitution of an associated word for another: it uses contiguity (Jakobson, 1999). Spatial or temporal adjacency, time, space and causal relationships are metonymic relationships. An example of a metonym is "He is fond of the bottle." There is a realistic connection here between a bottle and alcohol, because alcohol is usually stored in a bottle.

Metaphor: Metaphor is not a formal associative relationship, but since the context of this research is the expression of relationships in architectural design, it is very noteworthy to include it here alongside associations. Metaphor works by similarity (Jakobson, 1999). The use of metaphor transfers qualities with another concept, rather than associations, as is the purpose of metonymy. "Metaphor communicates by selection (a focus on the similarity between things) and metonymy by combination (a focus on the association in time and space between things)" (Berger, 1995: 88). Metaphors are based on comparisons, while metonyms are based on realistic connections. In real life, the two processes are mixed together. Metonymic and metaphoric thinking is fundamental to human cognition (Ortony, 1993). An example of a metaphor is "He is fishing for information." This phrase transfers the concept of fishing, where the qualities are waiting and hoping to catch something that is at the moment invisible, into a different domain.

3.3. THE DOCUMENT STRUCTURE

The document structure of a complex information structure consists of a collection of multi-media design documents that are used in the conceptual design phase. Documents in this collection initially do not have semantic relationships with each other. During the process of giving meaning to and making sense out of this collection relationships are created. In this section processes and techniques about relating elements of the semantic structure with documents are discussed.

3.3.1. Decomposition of design documents by content

The elements of a semantic structure describe documents in the document structure. These descriptions are represented as relationships between elements of the semantic structure and documents. However, these relationships do not offer any information on the importance of the concepts or conceptual relationships for the document, or to which portions of the document these apply. Furthermore, users may opt to simply ignore concepts which apply to only part of a document. As such, these relationships offer a quantitative rather than a qualitative valuation of the document. Instead, by allowing a concept to be related to only a portion of a document, many more concepts that better fit parts of documents can be associated with the appropriate document portions. This will make the documents inherently related by content.

We propose decomposing documents by content and integrating these decompositions into a complex information structure. This implies both defining the document's structure, identifying document components (and subcomponents), and augmenting the structure's relatedness with content information. The semantic relationships between the resulting components make the documents inherently related by content. There are some representational issues involved in the process of interpreting, breaking up, and relating documents.

The input to the document decomposition process is a number of documents. The output is an integrated structure of components and relationships. In between, documents are decomposed and these components related. This process requires the specification of the composition structure and relationships. Components (and their subcomponents) can be recognized as representations of concepts in the semantic structure. These components are automatically related in a compositional manner and through relationships to the concepts they represent. Components may overlap, resulting into subcomponents that belong to more than one component. Components may also be grouped into meta-components, creating component-to-component relationships corresponding to relationships between concepts.

A concept does not impose any particular representational decomposition on the document depicting this concept. Instead, different documents rely on different vocabularies that have their origin in the domains of the respective document. These vocabularies may overlap but, more often, they will offer alternative descriptions of related concepts reflecting on the context at hand. A structural approach to describing document decompositions and their integration into a single information structure is proposed, considering a self-similar, structural decomposition of a document into document components using the same representational syntax (Figure 3.13). The semantics of the decomposition is separately specified in the semantic structure. Separating semantics from syntax in the decomposition offers additional flexibility. Users can alter either the decomposition or the categorization without affecting the other. This approach to decomposing documents provides a uniform structure that is easily adaptable, unlike a semantic decomposition according to a product model[40]. As a result, by selecting the number of associated concepts and the level of decomposition, the user has full control on the effective positioning of her document within the information structure, irrespective of the document in the document hierarchy (see Figure 3.6).

Decomposing documents by content facilitates creating a complex information structure. Recognizing instances of concepts in documents within the information structure provides both qualitative and quantitative information about the importance of a concept for the document. It automatically augments the structure's relatedness by increasing both the number of information entities and their relationships. Replacing documents with component structures increases the number of information entities. Compositional relationships between document components extend the network of relationships. Relating concepts to document components allows for the specification of concepts that may otherwise be ill-suited to relate to the document at large. The resulting information structure allows one to access specific information directly through relationships to concepts, instead of requiring a traversal of the documents hierarchy. Individual components can be reached and retrieved more quickly when provided with more relationships.

[40] See footnote 24 about product models and ontologies.

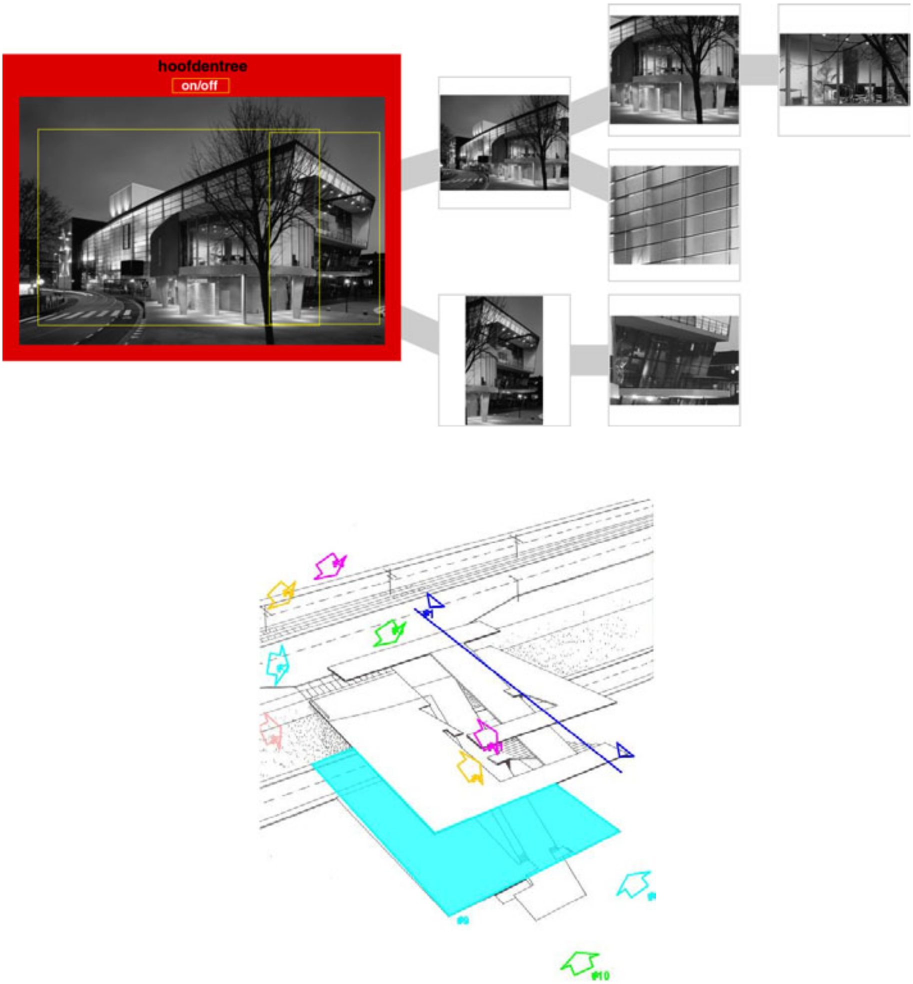


Figure 3.13. Two document decomposition examples. Top: The image has been decomposed by selecting rectangular areas and these areas behave as document components, linked to the original document with document relationships. Bottom: Markers have been placed on the document that denote a certain pixel location (or area) of the image for more specific indexing of those parts of the image. These locations are linked to the original document with document relationships. See Section 5.2 for more information about the content of this image.

3.3.2. Technologies for creating the document structure

A self-similar, structural decomposition of documents particularly applies to texts, images, and simple line drawings, as these lack strong inherent structure. All composed of symbols from a relatively small vocabulary, i.e., characters, pixels, and line segments, in simple one- and two-dimensional patterns, they are represented in a similar structure and can be operated on in a similar way: divided into smaller parts and the parts organized into a hierarchical structure. Other document formats such as more sophisticated geometric models may require a different representational language as well as a somewhat different approach to recognizing the relationships, as dependent on the format. Since a large part of design documents consist of 2-D drawings (projections), diagrams, photographs and text, this selection of formats is not an important limitation in the short term.

When dealing with texts, neural networks and pattern recognition algorithms can pinpoint keywords in and extract key concepts from documents (Greenberg, 1999a). Determining instances of concepts in a text is achieved by identifying content patterns associated to concepts and recognizing the same or similar patterns in the text. For simple line drawings, shape recognition algorithms can be based on the matching of distinguishable elements in the drawing and the concept descriptions (Krishnamurti and Earl, 1992; Krishnamurti and Stouffs, 1997). In order to automate the process of decomposing images, a four-step approach is proposed. Starting with a collection of concepts that may be represented in these images, it has been proceeded from the assumption that each concept has an associated set of shapes and forms dependent on the current context that makes it possible to recognize this concept within the images.

The first step is to determine the intrinsic structure (Barrow and Tenenbaum, 1981) of the scene, reflecting on the spatial properties of this scene. Using image processing and manipulation techniques, the appearance of objects is enhanced and objects' edges accentuated, thereby, providing preliminary object description data such as edges, surfaces, surface orientations and distances. This is done to reduce the large amount of information available in an image and to extract the useful information necessary for the next step. Neural networks can be used for the manipulation of image data.

The second step is to determine boundaries and regions of the geometry by segmenting and grouping the features in the intrinsic images. The resulting segmented images are formed by gathering the feature elements into sets likely to be associated with meaningful objects in the scene, i.e., edge segments corresponding to polyhedral edges. Some domain-dependent information may be used in this stage in order to determine the type of a boundary curve and to reduce noise. The form and shape information encoded within concepts plays an important role in providing this domain information.

The third step is the recovery of the geometry or shape of objects that make up the scene, from the line drawings resulting from the previous step. Information about regions and their adjacency, the relationships between boundary lines and vertices, and surface orientation information, enable the building of a geometric representation of the scene.

The last step is to interpret the geometry, matching object shapes with of the shapes and forms associated to concepts that may be in the scene. These matches must subsequently be controlled and validated. The overlaps between the geometries of matches can be optimized. The neighborhood relationships of these geometries can be validated by relying on the relationships of concepts within the semantic structure. Shape recognition and artificial intelligence techniques can further be used for the matching itself (Çiftçioglu et al., 1999), and for the control and validation of matches. As an example, neural networks are widely used for pattern recognition (Inoue and Urahama, 2000; Bishop, 1995).

While there has been a lot of research into the field of image and pattern recognition, especially in engineering, remarkably few practical applications of this research exist in the field of architecture. With the advances in web technologies, many institutions are placing their slide and image archives on the web (Gross, 1995; de Jong and Voordt, 2000). One can expect to have (semi-)automatic recognition mechanisms to be in place for the indexing of these images for effective and efficient retrieval (e.g., Restrepo, 2004). The functionality of such mechanisms in these environments should be pretty straightforward. It is to be expected that these technologies will mature and be able to serve this purpose. These technologies will surely provide a considerable benefit in the uptake of a system utilizing document decomposition.

3.4. CONCLUSION

In this chapter complex information structures, how to build them, their components, representation, and enabling technologies have been described. It was also delved in depth into processes and techniques that play an important role in the creation and maintenance of complex information structures. The ArcIMap framework is based on complex information structures.

Complex information structures have a high intensity and density in terms of amount and relatedness of information. This is crucial for their existence. In order to ensure this high intensity and density, the separation of the document structure and semantics, and decomposition of documents by content were discussed. Semantic and document structures and their relationships were explored. The process of building such a complex information structure within a community of practice must take work processes of the community into account; therefore, procedural issues about building up complex information structures are as important as technical and representational issues.

The following chapter formally introduces ArcIMap, its representation, its use cases in education and practice, exemplifies an educational context for it, and defines some guidelines for its applications.

4

ARCHITECTURAL INFORMATION MAP – ARCIMAP

Architectural Information Map (ArcIMap) is a framework consisting of a computational model[41] and a method for information organization and knowledge modeling in the conceptual phase of architectural design. Groups of designers who wish to store, organize, share, reason on and reuse collections of (visual) design information should use ArcIMap in order sustain their community of practice. ArcIMap defines a way to work with information collections without prescribing a fixed design vocabulary or classification structure. Users of an application of ArcIMap are also the creators of an information structure that resides in the application.

The goal of ArcIMap is to define a framework for the design and creation of digital applications that support designers in the conceptual phase of design, for example, by creating extensible libraries of design documents, and recording knowledge structures of designers. This framework can be used both in educational and professional contexts, although the process and requirements will differ. The final goal of the ArcIMap framework is to derive at specific implementations in specific contexts serving specific architectural communities of practice, yet from general principles (Tunçer and Stouffs, 1999). It is not the intention to develop a global system that can deal with all documents belonging to all kinds of building projects, but to define the representational framework for achieving an integrated information structure of components, relationships and matadata from a collection of design documents and the knowledge that resides in these documents. The framework can then be implemented for different purposes, domains, contexts, or architectural bodies.

ArcIMap supports designers to:

- individually or collectively organize their collections of visual material,

[41] A computational model is needed in order to create an information system. A computational model can be defined as a description of symbolic information and knowledge entities and their relationships that is an abstraction of reality in a certain context. A *model* according to the Oxford English Dictionary is: "A simplified or idealized description or conception of a particular system, situation, or process, often in mathematical terms, that is put forward as a basis for theoretical or empirical understanding, or for calculations, predictions, etc.; a conceptual or mental representation of something." Boman et al. (1997) state that "An effective approach to analyzing and understanding a complex phenomenon is to create a model of it. By a model is meant a simple and familiar structure or mechanism that can be used to interpret some part of reality. A model is always easier to study than the phenomenon it models, because it captures just a few of the aspects of the phenomenon". Schenck and Wilson (1994) define an *information model* as "a formal description of types of ideas, facts and processes which together form a model of a portion of interest of the real world and which provides an explicit set of interpretation rules."

- reuse the collected multi-media information in future projects,
- create semantic structures of concepts and relationships that reflect their design thinking and organizational processes,
- model domain knowledge,
- record personal design experiences and opinions,
- browse and search documents using the semantic structure of concepts and relationships,
- acquire, explore and share information,
- create a shared understanding and common language,
- correspond in a community of practice over knowledge and information stored in the system.

Implementations of the framework can support novice designers to learn about design solutions and expand their knowledge structure in a targeted way. Experienced designers ca n use the framework for organizing and storing visual material in a personal way. Project managers can use the framework for archiving collective material using a common information organization scheme. Groups of users can use the framework for directed communication.

The definition of ArcIMap (Figure 4.1) is based on the theoretical work on information and knowledge organization in conceptual architectural design and complex information structures described in Chapters 2 and 3, but as importantly, on the experiences learned from the prototype applications described in Chapter 5, and their evaluation results. These results provide a rich source for the iterative design process of ArcIMap, in accordance to *grounded theory* (Glaser and Strauss, 1967; Turner, 1983; Martin and Turner, 1986). Grounded theory states that through an evaluation of applications and their use, an information system gets grounded in its context, in an iterative process.

In this chapter, first some provisions for the definition of the ArcIMap framework are introduced followed by an informal description of the two components of the framework: the semantic and document structures. Next, the ArcIMap object model is formally described. In order to demonstrate a real-life use of the model, the description of the educational scenario previously introduced in chapter two as the aspiration for ArcIMap is demonstrated. This scenario has been further adapted to the structure and process defined by ArcIMap. Finally, a number of use cases are included in order to understand and explicate specific needs in the application design from the viewpoint of the users. The chapter ends with a number of guidelines for the design and implementation of an application of ArcIMap.

4.1. THE ARCIMAP FRAMEWORK

The two previous chapters have provided the underlying theoretical basis of ArcIMap from relevant fields of design and computation research. In this section, first, a number of provisions for the development of ArcIMap have been formulated. ArcIMap has been designed taking these as the starting point.

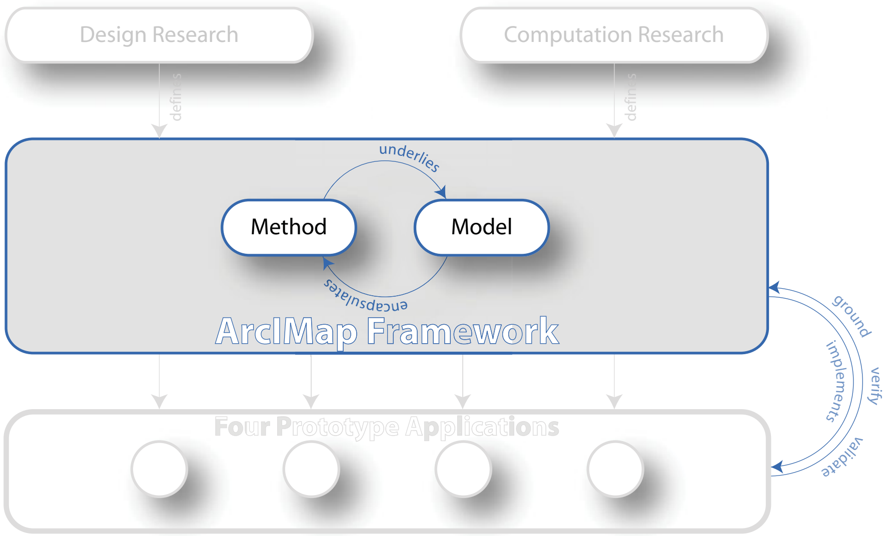


Figure 4.1. The ArcIMap framework encapsulates methods to describe social and information processes in order to be able to implement applications that are rooted in their context. ArcIMap also defines a computational model that acts as a structure for the design of complex adaptive information systems. Within the ArcIMap framework, the method underlies the model and the model encapsulates the method.

In the conceptual design phase, designers collect information that somehow relate to their design task, in order to gain information and knowledge, and to get inspiration. The very first requirement is that a framework for information and knowledge organization is needed that allows users to explicitly record and organize the knowledge that resides in the collected documents. This framework defines an open system in order to ensure reification (recording of common knowledge into documents) and participation in a community of practice. This system reflects on the process and context in which users operate.

Designers in the conceptual phase of design have difficulty specifying an exact query (Restrepo, 2004). If designers are supported to browse the information structure and to see the relationships between the entities contained in this structure, they can achieve a cognitive mode of browsing. In this cognitive associative mode of browsing, designers are able to recognize and follow connections between documents that arise from shared concepts and ideas in these documents. Therefore, the framework must implement an information structure that is sufficiently dense for allowing associative browsing: a complex information structure. A semantic structure made up of a network of concepts and semantic relationships acts as a backbone for the organization of knowledge and information within a complex information structure. A complex information structure contains, beside the semantic structure, a collection of documents called the document structure. Documents are interpreted and broken up into components, and these components within and between documents are related, and these relationships added to the representation. Elements of the semantic structure describe these documents or parts thereof. This is achieved in an information environment by defining elements of a semantic structure as metadata, e.g., as a keyword. This metadata is then assigned to documents or parts of documents. A complex information structure ensures the flexibility that is needed in considering the evolution of the shared knowledge within a community of practice.

In light of these provisions, in its main lines, ArcIMap has two components:

- **Semantic structure:** The semantic structure models *concepts* and *conceptual relationships* (knowledge), is conceptually derived from concept maps, is representationally a semantic network and additionally uses aspects from conceptual graphs and XML topic maps. The semantic structure acts as a backbone for the organization, decomposition and indexing of documents. Metadata derived from the content of documents together with metadata created independent of documents are represented as concepts in this structure. The concepts and conceptual relationships in the semantic structure are typed, respectively through the concept type hierarchy and the relationship type hierarchy.
  Conceptual relationships can be minimally typed according to a set of predefined (but extensible) semantic relationships in order to support information retrieval. When relationships are not typed, the danger looms that every relationship is unique, and retrieval result sets may be quite limited. Another advantage of typing the relationships is to make users consider the nature of the relationships they create. This will have a positive effect on the cognitive processes of users. In order to reduce the workload of users, the system should keep track of the labels used for types and suggest appropriate labels to users according to the selected semantic type. Figure 4.2 demonstrates the semantic structure in a simple example.

- **Document structure:** The document structure consists of a collection of multi-media design documents that are collected and produced during the conceptual design phase. These documents are named components in the model. Components can be decomposed further into (sub)components, and components can also be related to other components in other ways than component relationships. Document decomposition enables metadata to describe the parts of documents where they are most relevant, such that those parts of documents can be reached directly while searching or browsing. Occurrences relate components to members in the semantic structure. Occurrences are typed. Figure 4.3 demonstrates the document structure in a simple example.

Additionally, an application of ArcIMap is situated in a context of organizational and work processes of its users. The design of an application of ArcIMap must take the context in which this application will be used into account. An application of ArcIMap can only be successful when its usage and interaction principles suit the community of potential users. The work processes of the organization and users must be integrated in the design and interaction of the application. Therefore a participatory process is needed in the software and interaction design of the intended application. This requires a study of the users, their professional context, and a study of work and interaction processes in the form of interviews, observations and discussions within the use context of the application, and a translation of these into software requirements. Before embedding the application in its use context the users of the application must also be enlightened about the application, its purposes, and the goals and premises of the underlying model.

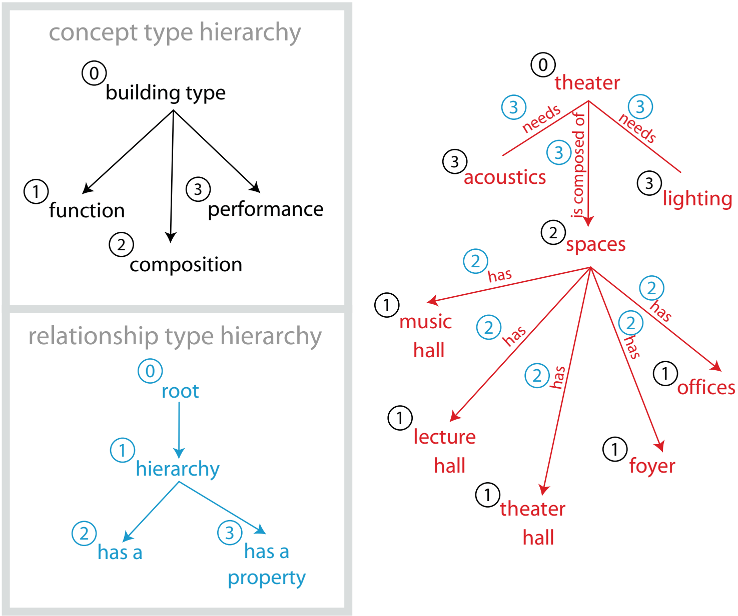


Figure 4.2. An exemplary diagram of a semantic structure of ArcIMap. The red structure is a hierarchy of concepts and conceptual relationships. The concepts and conceptual relationships are labeled. The black structure is a concept type hierarchy. Concepts have concept types denoted by the black numbers located next to them. The blue structure is the relationship type hierarchy. Conceptual relationships are typed denoted by the blue numbers located next to them.

The semantic structure of ArcIMap is a semantic network; however it makes use of properties of conceptual graphs and XML topic maps (see section 3.2.3). Below, some of the major similarities and differences between ArcIMap, conceptual graphs and XML topic maps are elaborated on.

Conceptual graphs: ArcIMap uses a number of ideas from conceptual graphs (see section 3.2.3 for a detailed description of conceptual graphs). The main differences between ArcIMap and the conceptual graphs representation are as follows:

- Conceptual graphs have been built upon predicate logic and are capable of formal knowledge representation. ArcIMap is not meant as a formal knowledge representation model.
- ArcIMap allows one concept to have various formalizations that can be used interchangeably in its representation. Conceptual graphs do not allow one concept to have a number of referents. In conceptual graphs, one needs to specify as many concepts as there are formalizations.
- In conceptual graphs, conceptual relationships can only be unidirectional. Bidirectional relationships are expressed as a pair of unidirectional conceptual relationships. ArcIMap allows for bidirectional conceptual relationships.

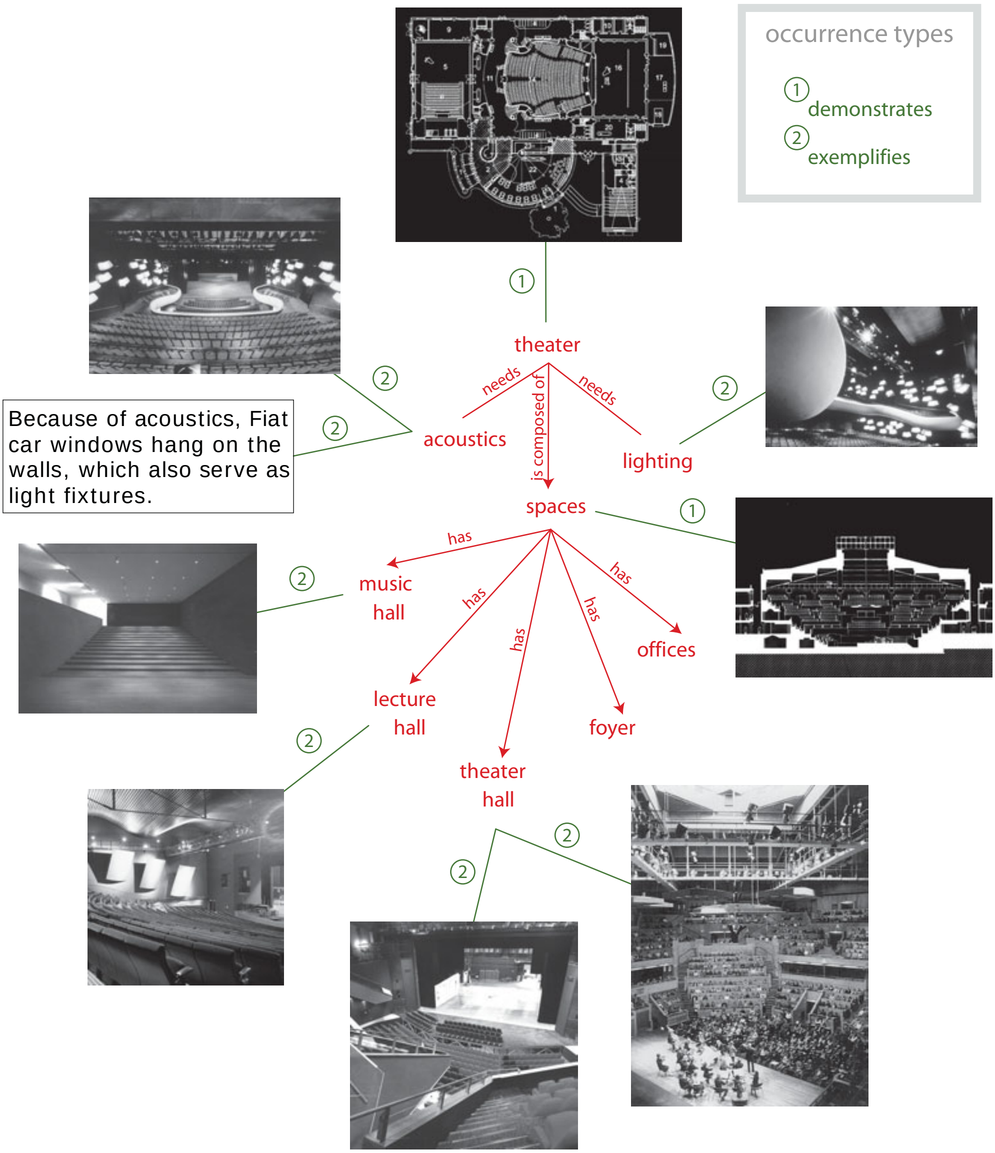


Figure 4.3. An exemplary diagram of a document structure of ArcIMap. The red structure is a hierarchy of concepts and conceptual relationships. The green list is a list of occurrence types. Occurrences are typed denoted by the green numbers located next to them.

- In conceptual graphs, concepts can be represented as conceptual graphs. This is not the case in ArcIMap. Such complex definitions are considered too complex for users to handle in the intended use of ArcIMap.
- In conceptual graphs, conceptual relationships are typed, as is the case in ArcIMap, but the relationships are labeled with their types. In ArcIMap on the other hand, the label and the type may be different. In a formal knowledge representation framework as conceptual graphs, the types of relationships are restricted. This does not allow for subjective relationships. The creation of subjective semantic relationships is one of the goals of ArcIMap.

XML Topic Maps: The general principles underlying topic maps and ArcIMap are quite similar (see section 3.2.3 for a detailed description of topic maps). Both utilize a conceptual semantic structure for information organization, and both use aspects from semantic networks and conceptual graphs. In both models, the semantic structure is separated from the information entities. The main differences between ArcIMap and XML topic maps are as follows:

- Topic maps are specifically designed for XML (although the underlying principles are universal).
- Topic maps are part of the semantic web development. Their purpose is to semantically access and organize the information on the web. ArcIMap is intended to model specific document spaces of which the scope and content is defined by the users.
- Topics allow for multiple names and these names have scopes, because topic maps have no defined context, and they potentially deal with all the information on the web. Multiple naming and scopes are a technique to support this. ArcIMap does have a context, namely the conceptual phase of design, therefore multiple names are not necessary. Synonym rings created through synonymy relationships is a more suitable technique for ArcIMap.
- Topic maps have only bidirectional relationships (associations) between topics, directional relationships are not included. This is semantically not powerful enough. ArcIMap allows for directional as well as bidirectional relationships between concepts.

4.2. FORMAL REPRESENTATIONAL STRUCTURE OF ARCIMAP

In this section, the structure of the ArcIMap model is formally introduced. The various objects that make up the model, and their relationships and interactions are described. These objects are also further elaborated with their most important attributes.

Formally, ArcIMap's semantic structure consists of a bipartite graph, a concept type inheritance hierarchy, and a relationship type inheritance hierarchy. The bipartite graph consists of *concepts* and *conceptual relationships* (Figure 4.4). A conceptual relationship relates two or more concepts.

A concept has a label that can be freely defined. A concept also has a (single) type that belongs to the concept type hierarchy. If a label is not specified for a concept, it is considered to be a generic instance of its type. A concept may also have various formalizations, for example, a sketch, a picture, etc. Furthermore, a concept has other attributes such as its creator and its creation date, and possibly an annotation that shortly explains the concept. The UML diagram in Figure 4.5 shows the concept as a class specification. In an implementation of the model, the concept class will have undoubtedly more attribute slots, but for reasons of abstraction and simplicity, these have not been further elaborated here. The same applies to all other objects described in this section.

Similarly, a conceptual relationship has a label that can be freely defined, and it also has a (single) type that belongs to the relation type hierarchy. If a label is not specified for a conceptual relationship, it is considered to be a generic instance of its relation type. A conceptual relation has a weight that denotes the importance of this relationship for the concepts it relates. It also has a boolean value that determines if this conceptual relationship is directional or not. The concepts that this relationship relates are represented as an ordered collection, named concepts. If the relationship is directional, then, only the last concept in the collection is at the receiving end of the relationship. Similar to a concept, a conceptual relationship may have various formalizations, an annotation, and must have a creator and a creation date. The UML diagram in Figure 4.6 shows the conceptual relationship as a class specification.

A concept type hierarchy relates the concept types using a generalization-specialization inheritance relationship. All members of this hierarchy must be unique. The UML diagram in Figure 4.7 shows how the concept object is related to the concept type hierarchy. A root concept type is always initially created. The properties of each type in the hierarchy are inherited by its subtypes. Concept types have a label, and they can be implemented with additional properties (Figure 4.8). A type hierarchy is usually created in its entirety in advance before creating the other parts of the knowledge structure. The way the type hierarchy is created depends on the use of the model and is elaborated on in the use cases section (section 4.4).

Similarly, a relationship type hierarchy relates relationship types using a generalization-specialization inheritance relationship. All members of this hierarchy must be unique. The UML diagram in Figure 4.9 shows how the conceptual relationship object is related to the relationship type hierarchy. A root relationship type is always initially created. The first level of the hierarchy may be composed of the semantic relationships that exist in the model, corresponding to the semantic relationships described in section 3.2.4. These semantic relationships can be freely specialized in lower branches of the hierarchy. The properties of each type in the hierarchy are inherited by its subtypes. Relationship types have a label and can be implemented with additional properties (Figure 4.10). The way the relationship type hierarchy is created depends on the use of the model and is elaborated on in the use cases section (section 4.4). However, the model suggests the first level of the relationship type hierarchy to be created in advance before creating the other parts of the knowledge structure, and this first level to be restricted.

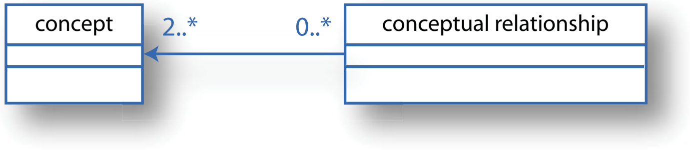


Figure 4.4. A UML class diagram describing the relationship between concept and conceptual relationship objects.

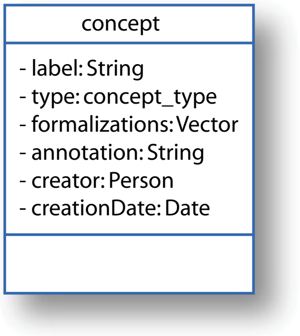


Figure 4.5. The concept object as a UML class diagram showing its main attributes and the data types of these attributes.

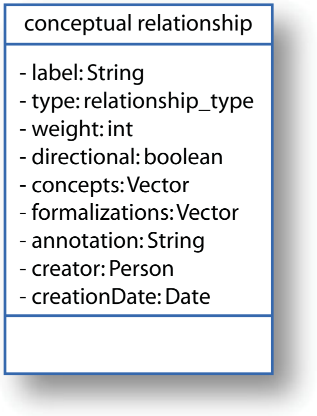


Figure 4.6. The conceptual relationship object as a UML class diagram showing its attributes and the data types of these attributes.

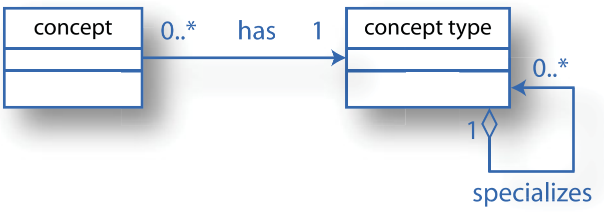


Figure 4.7. A UML class diagram describing the relationship between the concept and concept type objects.

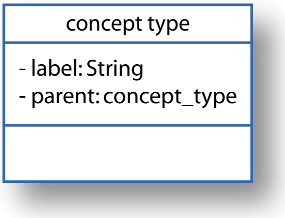


Figure 4.8. The concept type object as a UML class diagram showing its main attributes and the data types of these attributes.

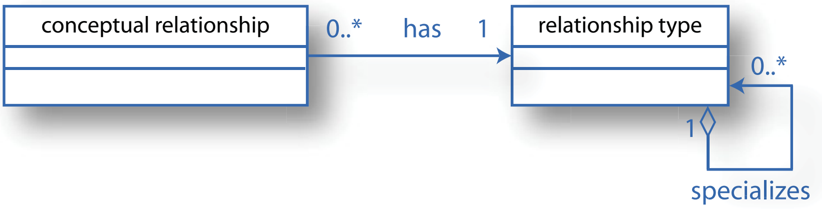


Figure 4.9. A UML class diagram describing the relationship between the conceptual relationship and relationship type objects.

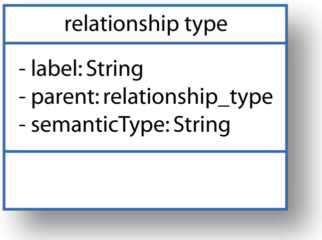


Figure 4.10. The relationship type object as a UML class diagram showing its main attributes and the data types of these attributes.

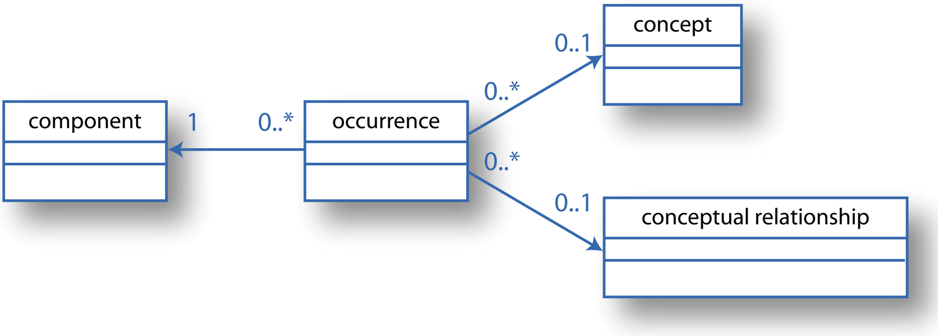


Figure 4.11. A UML class diagram describing the relationship between the component, occurrence, concept and conceptual relationship objects.

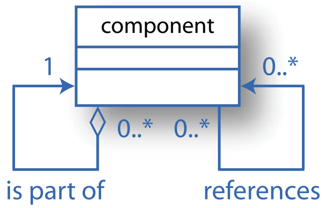


Figure 4.12. A UML class diagram describing the relationship between the component, occurrence, concept and conceptual relationship objects.

Documents in the system are represented by component objects. A component is related to a concept or a conceptual relationship by an occurrence object (Figure 4.11). The name occurrence has been chosen in coordination with the XML topic maps, where occurrences relate documents to topics, similar to their role in ArcIMap. Each occurrence object relates one component to either a concept or a conceptual relationship.

A component represents a document and its representation is independent of the document format in order to allow for a unified representation of all multi-media documents. A component can have two types of relationships to other components: a component can be a part of another component, and a component can reference one or more other components (Figure 4.12). The "is part of" (meronymy) relationship is an aggregation relationship; the "references" relationship is an association relationship. An aggregation relationship denotes that the whole (in this case: parent component) plays a more important role than its part (Rumbaugh et al., 2005). This relationship ensures that the representation of document decomposition independent of the syntax used in the document.

A component (Figure 4.13) object has a label, which could be considered as its title. If the user does not provide a label, the file name or URL can be used as the label. It has a format which determines how it would be displayed in an implementation. A component also has

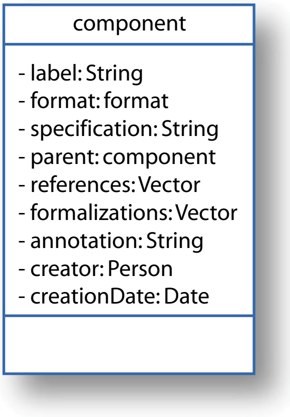


Figure 4.13. The component object as a UML class diagram showing its main attributes and the data types of these attributes.

a specification. The specification identifies the nature of the component, for example, a specification can be a file path, a URL, or if the component is part of another component, it can be a description of this part relationship, e.g., for an image component, the x and y coordinates and width and height values, for a part of a text the beginning and end character indexes. If the component defines a marker placed on another component, the specification value may consist of the kind of marker icon and the properties related to the positioning of the marker icon. These may be absolute values or they may be relative to the parent component. The specification property can be further extended to specific classes in an implementation of the model. A component may have a parent component, and/or a collection of references to other components. A component may also have a collection of formalizations. These can be used, for example, to present a text document with a visual thumbnail, or to present an image with another image as a thumbnail to distinguish itself from other results in a retrieval result set. A component may have an annotation, must have a creator and a creation date.

An occurrence relates a document component to a concept or conceptual relationship. An occurrence object (Figure 4.14) may have a label. An occurrence has a type that characterizes its nature. For example, a type may be "describes" or "exemplifies". The occurrence types are not organized in a hierarchy or any other structure, they are treated as a controlled vocabulary[42]. This ensures that all occurrence types are unique. Occurrence types classify occurrences into various classes of similar occurrences, whereas the labels are used for uniquely identifying an occurrence if desired. An occurrence has a reference to either a concept or a conceptual relationship, and one component. An occurrence has a weight that denotes the relevance of the component that it relates to a concept or conceptual relationship. An occurrence has a creator, a creation date, and can have an annotation.

[42] Please see Footnote #25.

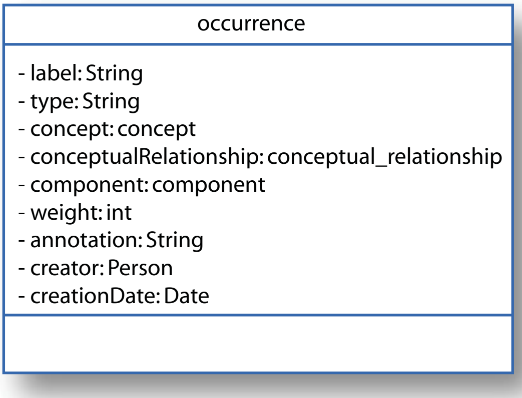


Figure 4.14. The occurrence object as a UML class diagram showing its main attributes and the data types of these attributes.

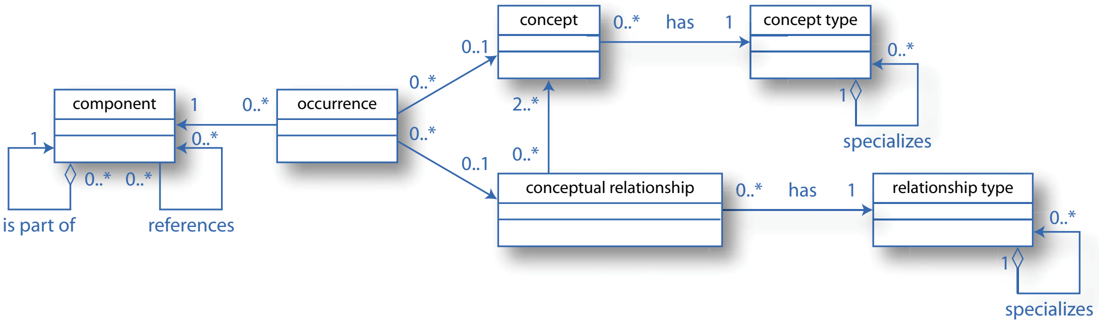


Figure 4.15. The ArcIMap model in its entirety as class objects and their relationships represented in UML.

Figure 4.15 shows the model in its entirety as class objects and their relationships.

4.3. A REAL-LIFE SCENARIO IN AN EDUCATIONAL CONTEXT

A collective information gathering and brainstorming workshop was developed within the context of a design exercise with students[43]. The design exercise was the redevelopment of an urban site in central Rotterdam (see Tunçer et al., 2005 for a detailed description of the

[43] An elective M.Sc. course that was offered at the Faculty of Architecture, Delft University of Technology, in the fall semester of 2004 to eight students. The name of the course was Mediated Discourse.

project). This workshop has been described in detail in Section 2.7. The goal of the first part of the workshop was to define In this workshop, students first created a concept map as a collective framework describing the important aspects that play a role in the preliminary design stage for this specific site, and the relationships between these aspects (see Figure 2.10). The next assignment of the students was to organize the documents they collected in the first part of the workshop using this concept map as the organizational structure. They were asked to assign concepts to each document they collected and to index and store the documents in the course database in that manner. The students were also allowed to create parts of documents and index them separately, and also to reference documents from other documents. This way, they collectively created an information structure that modeled their initial design concepts and their collective analysis of various aspects of the given site.

In this section, a part of this concept map and some of the associated documents and their relationships have been isolated and adapted to ArcIMap with the purpose of illustrating a real life example of the specification of the semantic structure and the document structure. The concepts in the concept map define concept objects, and the relationships between them define conceptual relationship objects. The documents define component objects. The relationships among documents are represented using the component relationships. The relationships between documents and concepts define occurrence objects. In order to adapt the concept map developed by students to ArcIMap, some additions were made:

- Definition of a concept type hierarchy
- Association of a concept type to each concept
- Extension of the basic relationship type hierarchy that comes with the model
- Association of a relationship type to each conceptual relationship
- Specification of a label to each conceptual relationship
- Definition of occurrence types
- Association of an occurrence type to each occurrence

An overall view of the information structure derived from this conversion is shown inFigure 4.16. In this specific example, no labels for occurrences were specified, only occurrence types were assigned to occurrences. Labels other than the assigned types were defined for some concepts and conceptual relationships.

Apart from the formal differences between the concept map and its adaptation to ArcIMap, a number of process related points also require attention. If the students were to use ArcIMap as the framework for this workshop, the workshop would need to be organized somewhat differently. Below, a number of considerations are listed that elaborate on this concern:

The definition of a concept network: According to the composition of the members of a group and their experience levels, and the cognitive load that is expected of them, a first level branching may be predefined in the definition of a concept network. Sometimes, this is necessary for a good start of a collective cognitive process (Davis et al., 2001). In the context of the course described here, the instructor of the course mediated the process and had an influence on the definition of the first level concepts, although this effect was not explicit.

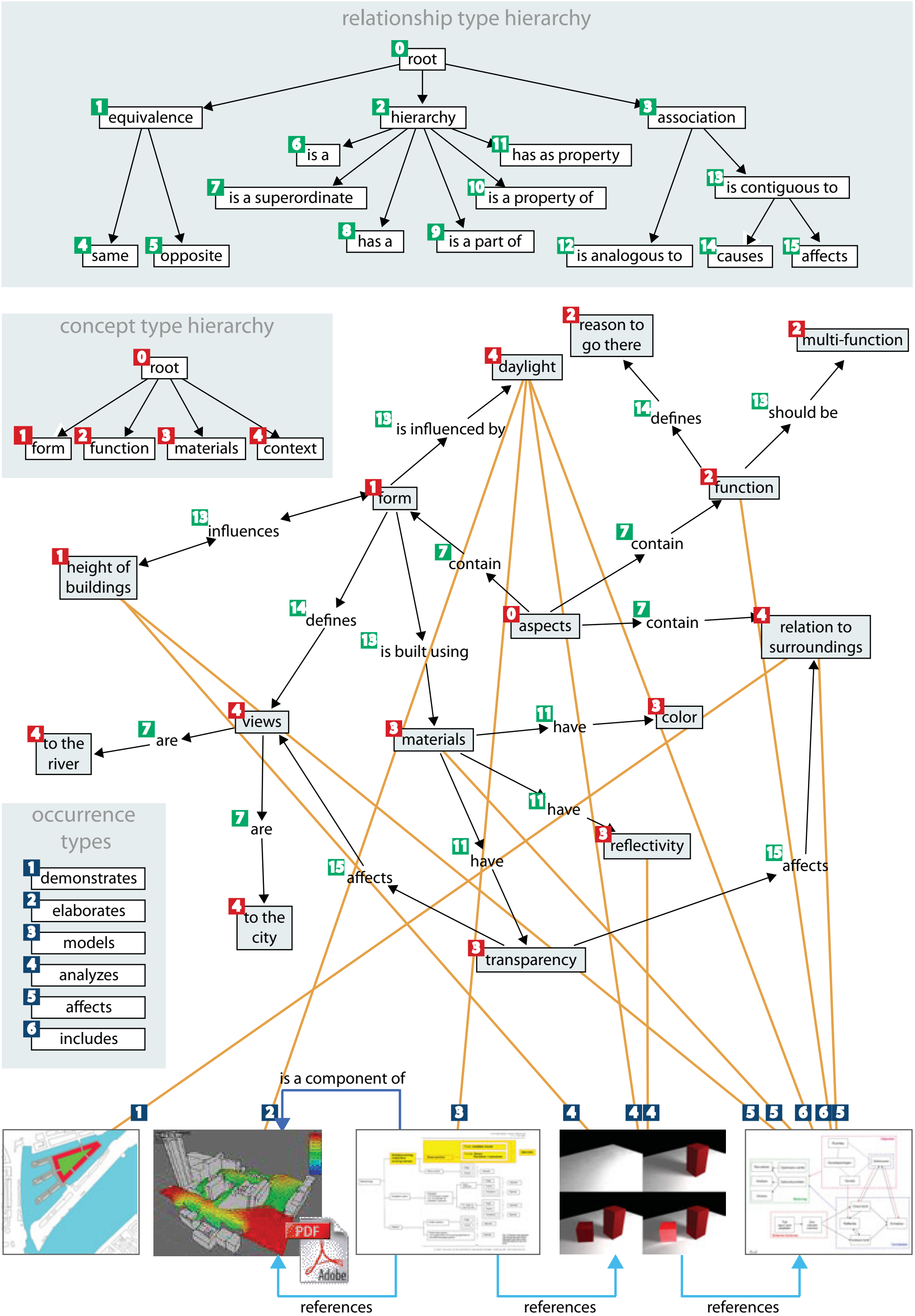


Figure 4.16. Diagram denoting the complex information structure derived from reification and participation processes of an educational community of practice. The information structure has a semantic structure consisting of typed concepts and typed conceptual relationships, a concept type hierarchy and a relationship type hierarchy, a document structure consisting of documents and components related through document relationships, and typed occurrences that relate elements of the semantic structure with elements of the document structure, and a list of occurrence types.

The definition of the concept type hierarchy: In such a course, the definition of the type hierarchy can either be done by the instructor ahead of time and given to the students in its entirety, possibly allowing the students to expand it, or students can build it up collaboratively as part of the brainstorming process. If students collaboratively define the type structure from scratch, this process needs to go hand in hand with the information gathering activities. Within the real process of the workshop, the students would have defined the type hierarchy right after a list of aspects was derived. The students would have grouped the aspects, and then abstracted the groups, and related the resulting types in a generalization-specialization hierarchy. Then, they would revise the aspects and relate them.

In any case, there should be an explanation session about the purpose of a type hierarchy. Students also need to be told that a type hierarchy needs to be general enough. Additionally, students need to know that they need to be careful when creating branches in a type hierarchy, because the relationships in a type hierarchy implement generalization-specialization principle. The 'children' of a type need to be somehow a specialization of the 'parent'.

The association of a concept type to each concept: A concept type denotes the nature of a concept. For example, 'views' is an aspect of 'context'. By concepts having concept types, the final concept network can be freed from unnecessary branching and redundancy. For example, if the concept types contain 'context' and this is associated with a concept in the network, the concept network does not need to have a specific 'context' concept, but only its instances. This frees up effort, time, and complexity. When students are doing the association process, they need to be aware of this. If students are creating the type hierarchy themselves, they need to hear this in the introductory explanation as well. Students also need to be aware of the fact that concept types serve as a link to other concepts, and this creates implicit relationships between concepts that are not specified explicitly.

Association of a relationship type to each conceptual relationship: The considerations for this point are similar to those for the point above, except that the first two levels of relationship types in the relationship type hierarchy are specified by the model. By associating a relationship type to a conceptual relationship, one explicitly defines the reason of existence of this relationship. This is very useful for people other than its creator to understand the nature of the composition of a concept network. Furthermore, when students explicitly think about relationships, this has a positive effect on their cognitive design thinking. This point should be included in the instruction to the students.

Specification of a label to each conceptual relationship: Specifying a label for a conceptual relationship is most useful when this is an associative relationship. An associative relationship can be highly subjective or personal, and therefore may need to be further specified than the rather generic relationship types. However, in general, the specification of a label is optional. When it is not specified, the label of its relationship type is taken over. These points need to be included in the instruction.

Definition of occurrence types and association of an occurrence type to each occurrence: Occurrence types are not related in a structure. The absence of a structure avoids the necessity to define occurrence types in advance, before the occurrences themselves are created. Instead, these types can be defined at the moment the occurrences are described. Their most important practical use is the grouping and retrieval of documents according to type. In an informal concept mapping session as described above, they may be omitted. But if students are modeling knowledge in a more formal way, comparable to the structure described in the BLIP prototype application described in section 5.2, their use is more essential in order to refine the role of the document explicitly for the concept or conceptual relationship.

The embedding of the system in the educational process: For the successful use in education of a system that implements ArcIMap, all the involved instructors must also actively use the system. Instructors must also retrieve and evaluate students' works as they have been submitted in the system. Students should not have to do double work for submission into the system on one hand and for submission to the instructor on the other hand. This ensures the motivation of the students.

4.4. USE CASES

ArcIMap enables the building up of a complex information structure. Complex information structures result from correspondence among members of architectural communities of practice. Therefore, ArcIMap applications must enable and support correspondence. These applications must possess mechanisms that don't change over time, but allow the content to change over time for the participants to be able to correspond on the content. The complexity of the information structure should not stand in the way of its ease of use, especially when integrating individual documents into it. Therefore, the tools, mechanisms, and techniques for creating the integrated information structure should be as clear, straightforward, and intuitive to use as possible. Additionally, the collective creation and maintenance of an organizational structure must be actively enabled by these mechanisms. This section starts by presenting some use related conditions for ArcIMap applications in order to fulfill these requirements.

An application of ArcIMap must support individual and group work. Users must have the ability to encode, search and extract design knowledge relevant to the problem at hand. In order to do so, users interact with the semantic and document structures of ArcIMap while operating in the embedded context. Such an application supports various information retrieval needs of the user concurrently: capturing, browsing, searching, and updating. In an ArcIMap application, examples are provided about the use of the application. Additionally, users are trained in advance about the use and purpose of the application.

While building a complex information structure, the semantic and document structures are concurrently adapted. The semantic structure guides the hierarchical document decomposition of multi-media documents, and the collected documents guide the expansion and modification of the semantic structure. The resulting semantic structure is project or institution specific. It can be created by a group process or can be imported as a ready made structure. The overall structure may also constitute a combination of networks, hierarchies and linear dependencies, describing different aspects or parts of a domain or knowledge structure. Users are able to create synonym rings by creating equivalence relationships between concepts in a semantic structure.

Users are able to associatively browse the information structure over conceptual relationships, component relationships, and associations. A graphical interaction mechanism that supports browsing needs to be presented to the users of ArcIMap applications. Search and find capabilities are presented to users in order to look for a specific document, look for documents related to a specific concept, etc.

After this introduction to use related informal requirements for ArcIMap applications, the rest of this section describes a number of use cases that have been worked out in detail. Earlier in this chapter the main objects that comprise ArcIMap have been presented. In software application design, the "use-case driven approach" takes the user's needs as the main criterion for software design. Jacobson et al. (1992) describe a method, named Object Oriented Software Engineering (OOSE), that advocates that use cases must be well thought out before going into specific application design (Cumming, 2005). The OOSE method uses "use cases" in order to understand and explicate specific needs in the application design from the viewpoint of the users. The use case model of OOSE has later been incorporated into UML (Dori, 2002: 404). Below, the methodology of OOSE is followed, using UML as the modeling language.

The described use cases determine the various scenarios and contexts of the use of the model and the interaction of users with the various components of the model. These use cases are not exhaustive. Two major use contexts are considered for an application of the model: educational, meaning in architectural education, and practical, meaning in an architectural office.

4.4.1. The specification of the semantic structure

This first use case describes the specification of the semantic structure of ArcIMap (Figure 4.17). Consider a group of users and the actions that these users can do:

- Create a relationship type
- Create a concept type
- Create a concept
- Create a conceptual relationship
- View the relationship type hierarchy
- View the concept type hierarchy
- View the concept network

The order of these actions depends on the specialization of this general use case, and this specialization is dependent on a context. There are also various actors to be distinguished in each context. Next, a number of specializations for this use case are described.

Defining a concept type hierarchy in an educational context: Actors in this use case are instructors and students. This use case can be further divided into two (also see section 4.3):

Case A. Use in bachelor curriculum: The students do not yet have much knowledge and experience, or a high level of responsibility (Stouffs et al., 2002), therefore the actors for this use case are instructors.

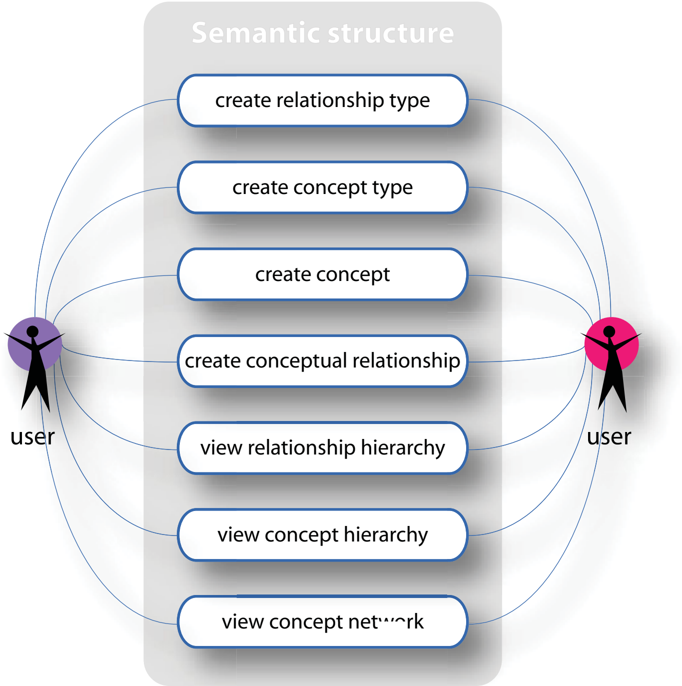


Figure 4.17. A UML use case diagram of actions that users can perform in order to create the semantic structure of ArcIMap.

Case B. Use in master's curriculum: The students are expected to perform independently and have a higher level of responsibility. Therefore, the actors in this use case are both the instructors and students.

The flow of events in this use case is:

1. *Case A.* The instructor (or a group of instructors) creates a collection of vocabulary terms (a controlled vocabulary) that are crucial for the content of the course. The instructor has in mind an archiving structure for the content that she expects the students to produce during the course. The instructor relates these vocabulary elements in a generalization-specification hierarchy.

2. *Case B.* The instructor does this action together with a group of students.

The object involved in this use case is *concept type*.

Defining a concept type hierarchy in a practical context: Actors in this use case are project managers, other managers, or experienced designers in a manager position.

The flow of events in this use case is:

1. The manager (or a group of managers) creates a collection of vocabulary terms (a controlled vocabulary) that are crucial for a categorization of the material produced and used at the office. The manager has in mind an archiving structure for the content produced in the office. The manager relates these vocabulary elements in a generalization-specification hierarchy.

The object involved in this use case is *concept type*.

Defining a relationship type hierarchy in an educational or practical context: Actors in this use case are instructors and managers.

The flow of events in this use case is:

1. The instructor(s) or manager(s) creates the first two levels of the relationship type (see section 3.2.4) hierarchy. The model already provides a template for this, defining the following types for the first two levels of the hierarchy:

   - Equivalence
     - same (synonym)
     - opposite (antonym)
   - Hierarchy[44]
     - is a (hyponym)
     - is a superordinate (hypernym)
     - has a (holonym)
     - is a part of (meronym)
     - is a property of
     - has as property
   - Association
     - is contiguous to (metonym)
     - is analogous to (metaphor)

The instructor or manager can also define a third level of relationship types if desired, placing the new types in a hierarchical generalization-specialization hierarchy under any type in the first level of the hierarchy.

The object involved in this use case is *relationship type*.

Creating a concept: Actors in this use case are, in principle, any user except a viewer. The access rights of the user determine if the user is allowed to create a concept.

The flow of events in this use case is:

[44] The 'hypernym', 'meronym' and 'has as property' relationship types can in fact be derived from, reprectively, 'hyponym', 'holonym' and 'is a property of' relationships. The reason they are included in the relationship type hierarchy is for reasons of usability. The users will have more flexibility in terms of selecting the direction of their relationships when assigning types to them when these relationships are included in the hierarchy.

1. The user selects “create a concept” through interacting with the system
2. The user browses through the concept type hierarchy
3. In case the user does not find an appropriate type for this concept, the user adds a new type to the type hierarchy (if the user’s access rights allow for this action)
4. The user selects one type and associates it with the new concept
5. The user defines a label for the concept by typing one or a small set of words or the label of the concept type becomes the label for the new concept
6. If desired, the user uploads a set of formalizations for the new concept
7. If desired, the user attaches a short annotation to the new concept describing the meaning of the concept in the given context
8. the user selects the “create” option
9. The system automatically records the creator and the creation date of the new concept

The objects involved in this use case are *concept* and *concept type*.

Creating a conceptual relationship: Actors in this use case are, in principle, any user except a viewer. The access rights of the user determine if the user is allowed to create a conceptual relationship.

1. The user views the concept network and considers which concepts the intended relationship will relate
2. The user selects “create a conceptual relationship” option
3. The user selects one or more concepts as the starting concepts of the conceptual relationship
4. The user selects one concept as the ending concept of the conceptual relationship
5. If the user wishes to create a bidirectional relationship instead of a directional one, the user selects the option to make this relationship a bidirectional one
6. The user browses through the relationship type hierarchy
7. In case the user does not find an appropriate relationship type for this concept, the user adds a new type to the relationship type hierarchy (if the user’s access rights allow for this action)
8. The user selects one relationship type and associates it with the new conceptual relationship
9. The user defines a label for the conceptual relationship by typing one or a small set of words or the label of the relationship type becomes the label for the new conceptual relationship

10. If desired, the user may attach a weight to the conceptual relationship, which is a number in a range selected by the application (for example between 0 and 1)
11. If desired, the user uploads a set of formalizations for the new conceptual relationship
12. If desired, the user attaches a short annotation to the new conceptual relationship describing the reason this relationship exists in the given context
13. The user selects the "create" option
14. The system automatically records the creator and the creation date of the new conceptual relationship

The objects involved in this use case are *concept*, *conceptual relationship* and *relationship type*.

4.4.2. The specification of the document structure

This second case describes the specification of the document structure of ArcIMap (Figure 4.18). Consider a group of users and the actions that these users can do:

- Create a file upload component
- Create a URL component
- Create a marker component
- Assign component as an occurrence to a concept
- Assign component as an occurrence to a conceptual relationship
- View the concept network
- View the occurrences of a particular concept or conceptual relationship
- View occurrences
- Add a component as a reference to another component

When describing the use case for creating the document structure, neither the context differences of the educational and practical settings, nor the place of the actor in the organizational hierarchy play an important role. Next, a number of specializations for this use case are described.

Uploading a file component: The flow of events in this use case is:

1. The user selects "upload a file component" through interacting with the system
2. If desired, the user specifies a title for the component to be created
3. The user specifies the location of the file to be uploaded
4. If desired, the user specifies a thumbnail (image) location
5. If desired, the user specifies a short annotation describing information about this component

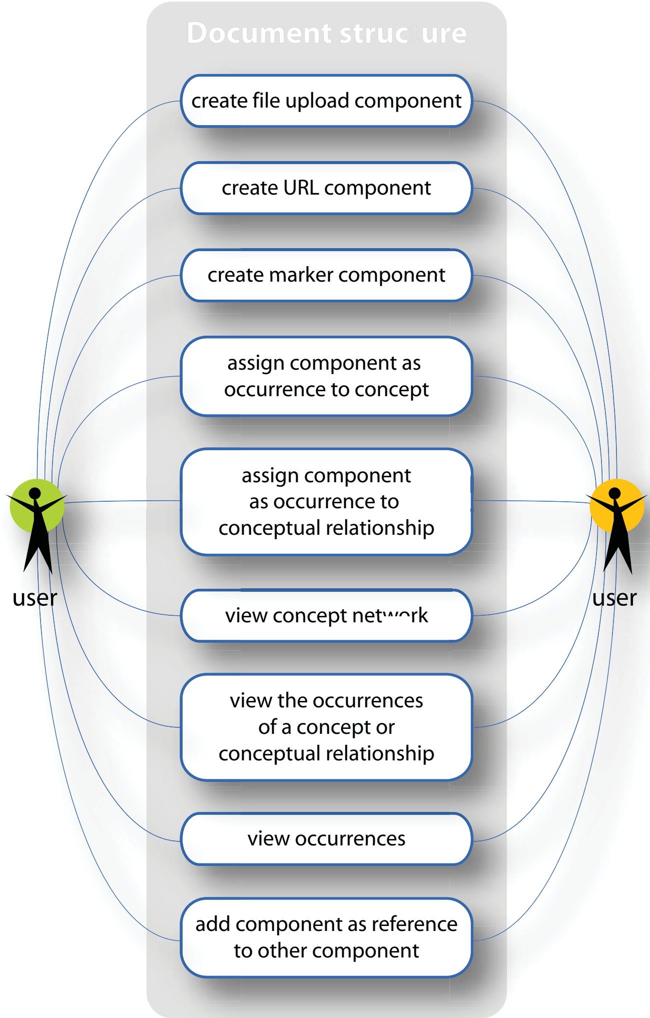


Figure 4.18. A UML use case diagram of actions that users can perform in order to create the document structure of ArcIMap.

6. The user selects the upload option
7. System automatically records the creator and the creation date of the new component

The object involved in this use case is *component*.

Creating a URL component: The flow of events in this use case is the same as the previous one, except for steps 2 and 5:

2. The user specifies the URL of the site
5. The user selects the create option

The object involved in this use case is *component*.

Creating a marker component: This use case has been demonstrated in 5.2. It may be used, among others, for analysis purposes.

The flow of events in this use case is:

1. The user selects an image component
2. The user selects a visual marker
3. The user places the marker on the component
4. If desired, the user specifies an annotation for the marker for describing the purpose and meaning of the marker on the component
5. The user selects the create option
6. The action of placing a marker on a component specifies a new component and the specification properties of the marker (location coordinates, etc.) are stored as part the new component. The new component is related to the first component with an "is part of" relationship.

The object involved in this use case is *component*.

Adding a component as a reference to another component: The flow of events in this use case is:

1. The user selects a marker placed on an image component
2. The user selects one or more other image components
3. The user selects "relate components to marker" option in order to relate these other components to the marker. This action relates these components to the selected marker component with the "references" relationship.

The object involved in this use case is *component*.

Assigning a component as an occurrence to a concept or conceptual relationship: The flow of events in this use case is:

1. The user selects a component
2. The user views the concept network

3. The user selects a concept or conceptual relationship to associate with the selected component with an occurrence relationship
4. If desired, the user views the occurrences of the selected concept or conceptual relationship
5. The user links the component to the concept or conceptual relationship by interacting with the user interface, whereby an occurrence is created
6. The user views the occurrence types
7. In case the user does not find an appropriate occurrence type for this occurrence, the user adds a new type to the occurrence types (if the user's access rights allow for this action)
8. User selects one occurrence type and associates it with the new occurrence
9. If desired, the user views the collection of occurrences in the system to get ideas about possible occurrence labels and other properties
10. The user defines a label for the occurrence by typing one or a small set of words or the label of the occurrence type becomes the label for the created occurrence
11. If desired, the user may attach a weight to the occurrence, which is a number in a range selected by the application (for example between 0 and 1)
12. If desired, the user attaches a short annotation to the new occurrence describing the reason this occurrence exists in the given context
13. The user selects the "create" option
14. The system automatically records the creator and the creation date of the new occurrence

The objects involved in this use case are *component*, *occurrence*, and *concept* or *conceptual relationship*.

4.4.3. Automatically filling the system from a file structure

This third use case describes the automatic transfer of documents stored in a file system into an implemented system of ArcIMap (Figure 4.19). The actor in this use case is a program, namely, a crawler, and the actions for this actor are (in the order given below):

- Generate concept types from directory names
- Generate concepts corresponding to concept types
- Generate conceptual relationships from directory structure
- Generate components from files
- Relate components to matching concepts

The assumption in this use case is that the directory names specify a classification structure (see section 5.4). This classification structure forms the initial semantic structure or is integrated in the semantic structure of the system. This use case is specifically suited for creating an image library in an educational or practical context. The communication and

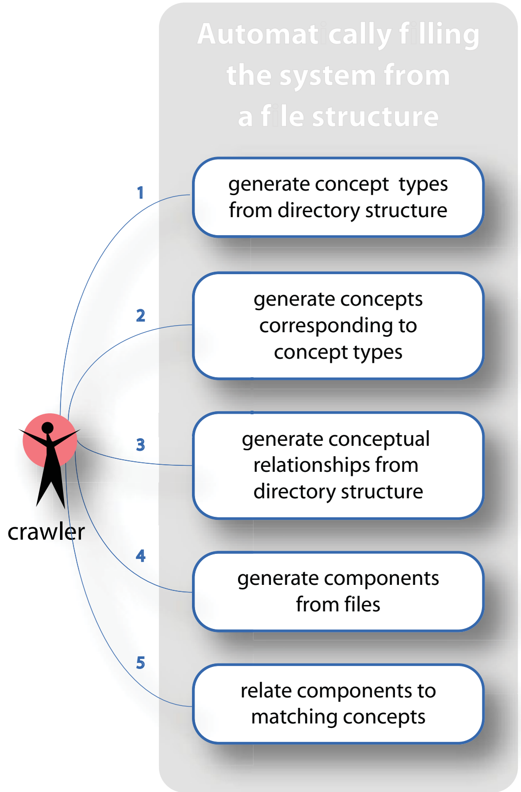


Figure 4.19. A UML use case diagram of actions that a crawler performs in order to transfer documents and their structure into an implemented application of ArcIMap.

correspondence aspect is of less importance in this use case. Users can and should go over the generated content and adjust it to the needs of the context if necessary.

Another assumption is that the first level of the relationship type hierarchy has been created according to the model suggestion.

This crawler program is run by an actor who has sufficient access rights. This will generally be the system administrator, a manager in a practical context, or an instructor in an educational context. The objects that are created by the crawler have as creator "crawler", but a log in the system keeps track of which user ran the crawler and when. This will allow users to trace which user has generated the objects.

Next, two specializations for this use case are described.

Defining the concept type hierarchy: The flow of events in this use case is:

1. Generate a temporary hierarchy of types from directory names using the nesting of the directory structure
2. Compare this structure with the existing concept type hierarchy in the system
3. If the type hierarchies match completely, do not add anything to the concept type hierarchy
4. If none of the members of the temporary type hierarchy exist in the concept type hierarchy, add the temporary type hierarchy as a new branch to the root of the concept type hierarchy
5. If all members of the temporary type hierarchy that exist in the concept type hierarchy define a sub-hierarchy that completely matches part of the temporary type hierarchy, then add the missing members to the concept type hierarchy according to the relationships in the temporary type hierarchy
6. Otherwise, provide feedback to the user on the discrepancies between the temporary type hierarchy and the concept type hierarchy

The object involved in this use case is *concept type*.

Defining the rest of the semantic structure and the document structure: The flow of events in this use case is:

1. In a recursive manner, generate concepts corresponding to the concept type hierarchy: for each concept type: check the collection of concepts in the system for concepts associated to the selected concept type without labels. If this exists, do not create a new concept, otherwise create a new concept associated with the selected concept type without a label.
2. Generate conceptual relationships between concepts corresponding to the relationships in the concept type hierarchy. These relationships are all hierarchical, and the relationship types are “is a” or “is a part of”. These relationships are not labeled.
3. Generate components from files
4. Relate components to matching concepts by creating occurrences. The occurrences should all have the same occurrence type and no label. The occurrence type chosen may be “describes”.

The objects involved in this use case are *concept type*, *concept*, *relationship type*, *conceptual relationship*, *component*, and *occurrence*.

Additionally, a mapping mechanism for unifying semantic structures that filters all the concept networks and fits them into a mediated archival classification system should be present in an ArcIMap application. This mechanism will work in the same way as described in this use case, with the difference that concept and concept type names will be read and matched existing structures instead of a directory structure.

4.5. GUIDELINES FOR ARCIMAP APPLICATIONS

An application of ArcIMap needs to be rooted in its operational context in order to have a successful use. Therefore specific use cases need to be fleshed out for specific applications during their design process. However, there are also some other informal requirements for these applications. These are described in this section as a preparation for the next chapter where four prototype applications of ArcIMap that were developed within this research have been described.

An application of ArcIMap must have a user friendly user interface. 'User friendliness' means in this context that the user interface is simple, easy to use, intuitive to interact with, and efficient. For instance, the interface may incorporate drag-drop mechanisms for uploading multiple documents, for relating concepts or semantic relationships to document components, etc. The interface should offer multiple, simultaneous views to enable "creative coincidences" by, e.g., viewing multiple documents at once using a post-it note like interface. The interface does not necessarily display all document attributes such as document title and annotation. This depends on the specifics of the context.

An application of ArcIMap must have a visually oriented user interface. The interface should offer visual accessibility to a large amount of information with their dependencies. Access to multi-media content (documents and concepts) must be included in the interface environment. The interface should contain detailed views and overviews. Overviews should have filtering functionalities. The interface should allow for easily switching between detailed views and overviews. In order to give users the possibility to keep track of their browsing process, the interface should visualize part of the browsing history.

Visualizing the semantic structure of an ArcIMap application facilitates an effective use of this structure in the process of constructing and manipulating this structure. Possibilities for an 'at-a-glance' overview and easy browsing are highly recommended in order to alter the structure effectively and easily. Therefore, visual, and possibly dynamic, displays for the semantic structures are highly recommended. Eloquent, effective and dynamic information visualizations should be present in the interface for relating concepts to documents, adding new concepts, etc. The interface should reflect on all properties of the semantic structure such as association of document components to conceptual relationships, directionality of conceptual relationships, and relationship types. The interface should distinguish various kinds of semantic relationships: relationships of equivalence, relationships of hierarchy, and relationships of association. Additionally, while constructing or expanding the semantic structure, the interface should allow users to temporarily create 'loose' concepts; i.e., concepts without any conceptual relationships.

The interface should allow for personalization, e.g., by showings all or some of the various formalizations of concepts as a user preference. In addition to the semantic structure, an application of ArcIMap should always offer some general classification of information entities, such as "by architect", "by building", "by building type", etc.

Since an application of ArcIMap supports communities of practice, it supports multiple users accessing the same information. Therefore, it is appropriate to presume that it should ensure independency of time and physical space in order to support remote collaborative work. This will be ensured by designing and implementing the application as a web-based system. Such an application should be synchronized and reflect all changes immediately on all users' applications. An information exchange protocol must be considered for the application to communicate with the external world. XML is an appropriate choice for this purpose.

An application of ArcIMap should support the specification and maintenance of access rights in order to allow for effective collaborative work. These access rights will reflect on the responsibility levels of users of the applications.

4.6. CONCLUSION

In this chapter the ArcIMap framework has been formally described. ArcIMap has been iteratively derived by using the knowledge gained from the study presented in the previous chapters and the experiences gained from the use and evaluations of the applications presented in the next chapter. ArcIMap defines a framework for applications that are implemented using it; however, in order to be successful in its use, any application of ArcIMap must be designed and built considering its use context and users.

The next chapter presents four applications of ArcIMap, rooted in architectural education and practice, where each application implements and tests certain aspects of ArcIMap. These applications have value on their own as information systems, and they have also provided feedback to the (re)development of the model.

5

APPLICATIONS OF ARCIMAP IN ARCHITECTURAL EDUCATION AND PRACTICE

Architectural Information Map (ArcIMap) is a framework consisting of a method and model that can be implemented into applications in order to support the conceptual design phase of architectural design. ArcIMap defines a structure for the design and creation of digital applications for storing, organizing, communicating, sharing, and reusing architectural information and knowledge. Applications of ArcIMap can create complex adaptive information systems as the means for (and the result of) architectural communities of practice to correspond on digital design information and knowledge during the conceptual phase of architectural design. Using an application of ArcIMap is a way of performing design research that visualizes the common processes produced by a design team. Such applications use state-of-the-art technologies without forcing a fixed way of decision making on their users.

The final goal of ArcIMap is to derive at specific implementations, yet from general principles (Tunçer and Stouffs, 1999). It is not the intention to develop a global system that can deal with all documents belonging to all kinds of building projects, but to define the representational framework for achieving an integrated information structure of components, relationships and metadata from a collection of design documents and the knowledge that resides in these documents. The framework can then be implemented for different purposes, domains, contexts, or architectural bodies. An application of ArcIMap must be rooted in its use context, therefore, a study of the social and work processes of the users and the organizational structure of the context in which it will be used must be studied in the design stage of the application. Additionally, its users must receive instruction on the concepts behind ArcIMap before using the system.

Four prototype applications of ArcIMap have been implemented in order to research, evaluate and verify aspects of ArcIMap as a method and a model (Figure 5.1). Some of these aspects are technical, usability, and embedding in the considered context. These applications supply valuable feedback to ArcIMap through the experiences gained and the evaluation results. Within the context of *grounded theory* (Glaser and Strauss, 1967; Martin and Turner, 1986; Turner, 1983), an information system gets grounded in its context in an iterative process through an evaluation of its applications and their use.

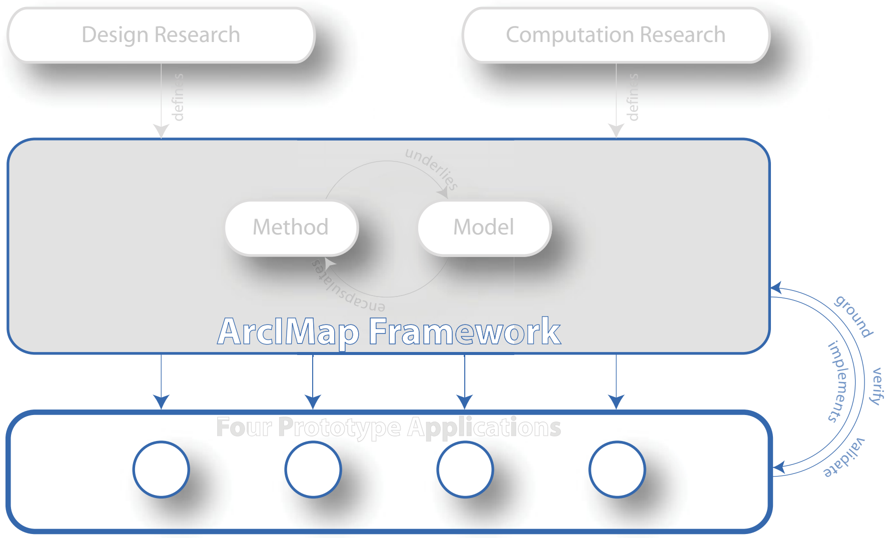


Figure 5.1. Four prototype applications are implemented using the ArcIMap framework in order to research, evaluate and verify aspects of ArcIMap as a method and a model. These applications ground, verify and validate the framework.

The prototype applications of ArcIMap are both educational and practical. An *Analysis Presentation Tool* that uses three Ottoman mosques as its case study, researches and validates the framework's unified representation, and evaluates the notions of interaction and associative browsing in an application of ArcIMap. *Blob Inventory Project (BLIP)* is a precedent library designed for modeling knowledge that has emerged from digital design, engineering and production processes of free-form geometry buildings and has been used in the 3rd semester of the M.Sc. architecture education. BLIP researches and evaluates user interface and interaction aspects of ArcIMap. *Design Analysis Network (DAN)* is a complex adaptive information system implemented as an educational architectural analysis environment used in the undergraduate 2nd year design studio. DAN researches and evaluates all components of ArcIMap, but especially the embedding and situating of the application within its context. *DesignMap* is a flexible and extensible content management system intended to be used at the early stages of design, is targeted towards small and medium-sized architectural offices, and has been used and evaluated at the architectural office Mecanoo in Delft. DesignMap researches, implements and evaluates ArcIMap within the context of architectural practice.

5.1. ANALYSIS PRESENTATION TOOL – THREE OTTOMAN MOSQUES: A PROOF OF CONCEPT OF ARCIMAP

As a proof of concept of ArcIMap, a prototype application in the form of a web-based environment with the purpose of building up, storing, and presenting architectural analyses has been developed. This prototype application has not been actively used, but acts as a test case for implementing and studying several aspects of ArcIMap. This application is predominantly concerned with building up a complex information structure, and interacting with it. As such, it takes the technical, interaction and interface issues into account. As the complex information structure was created by one person, the application's aim has never been sustaining a complex adaptive system of a community of practice.

Analysis plays an important role in architectural design and education. From a representational point of view, an analysis is composed of various abstractions, or design documents. An information structure that integrates the different aspects of the analysis, such that the analysis can be interpreted and used in ways other than the original abstractions present, would be particularly useful in education.

The analysis presentation tool presents a decomposition of abstractions by content using a semantic network. This network offers a flexible and extensible categorization structure for the abstractions in order to allow for associative browsing of the content. The concept network is specifically defined corresponding to the subject of the analysis. As a result of the separation of the document structure and the semantic structure, the construction of this network can easily be altered even after abstractions have been decomposed.

Ottoman mosques serve as a subject matter for this application. Three mosques by the same architect, Sinan (1490-1588), that present three different typologies of classical Ottoman architecture in their spatial and structural characteristics have been selected (Figure 5.2). These mosques are Şehzade (İstanbul), Süleymaniye (İstanbul), and Selimiye (Edirne). Ottoman mosques of the classical period are a good typological choice for this application, because:

- The architectural domain has a large enough body of documented and/or built examples.
- There is a sufficient amount of sources, research, and access to knowledge about the body of work.
- There is variety and evolution in the architectural domain throughout the time of consideration.
- The aspects taken into account are not limited to only one view (e.g. only floor plans), but include other attributes (e.g. cultural value, esthetic, structural, geometric, etc.). The body of work is influential enough during its period of life to be able to accommodate all these aspects.
- The body of work chosen is of interest to professionals such as architects and art historians.

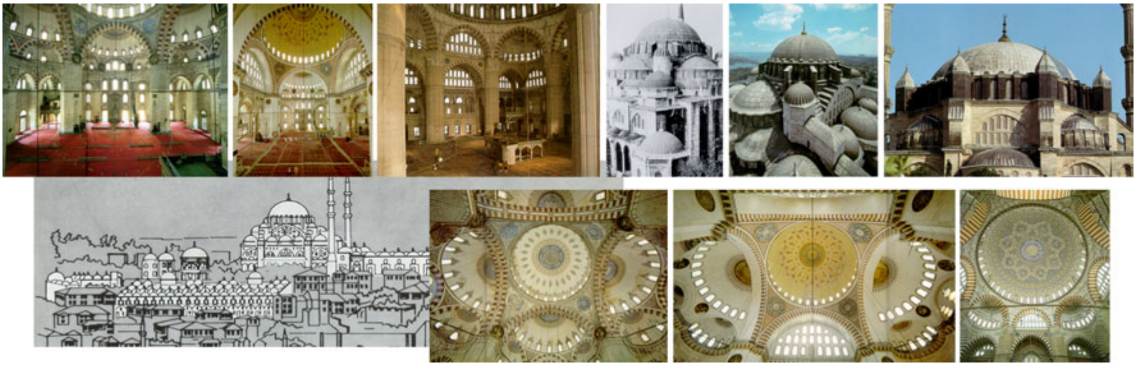


Figure 5.2. Sets of images from three representative mosques, Şehzade (Istanbul), Süleymaniye (Istanbul), and Selimiye (Edirne): interior space, dome structure as seen from the outside, silhouette, and central dome(s). Images from Egli (1997), Stierlin (1985; , 1998), and Erzen (1996).

The input to the application is a set of design documents in the form of images, texts and simple line drawings, and a semantic structure of concepts. The semantic structure is derived from the typology of classical period Ottoman mosques with the concepts represented as keywords. The explicit linking between documents and concepts in the typology is achieved simply through assignment. In this typology, the semantic relationships between the concepts are of only one kind: hierarchical. No explicit distinction is made between hyponymy and meronymy relationships (see section 3.2.4).

The output of the application is an integrated information structure of components and relationships. Access to the information structure is provided from a network of concepts and through the collection of documents. Documents are decomposed into their constituent components defining the document structure, in correspondence to the adopted concept network. Each component is assigned at least one concept.

5.1.1. Types and typologies

The network of concepts in this application is derived from the typology of classical period Ottoman mosque architectural style (Figure 5.3). This typology is the result of research of the architectural works and their components and the historical and cultural context (Egli, 1997; Erzen, 1996; Stierlin, 1998; Stierlin, 1985; Ertug, 1981; Kuban, 1987; Kuban, 1980). Figure 5.4 shows some types and instances in the context of Ottoman mosques.

Typology is the study or systematic classification of types. Within a discipline, members commonly share a definition and classification of common concepts. This structuring of shared knowledge through common concepts gives insight into that discipline (Leupen et al., 1997). Architects generally classify building designs based on spatial and formal features. This classification features the concepts of type and typology. For example, we can define museums, offices, or libraries as building types. Types and typologies are extensively utilized in architectural education and practice in spite of the controversy

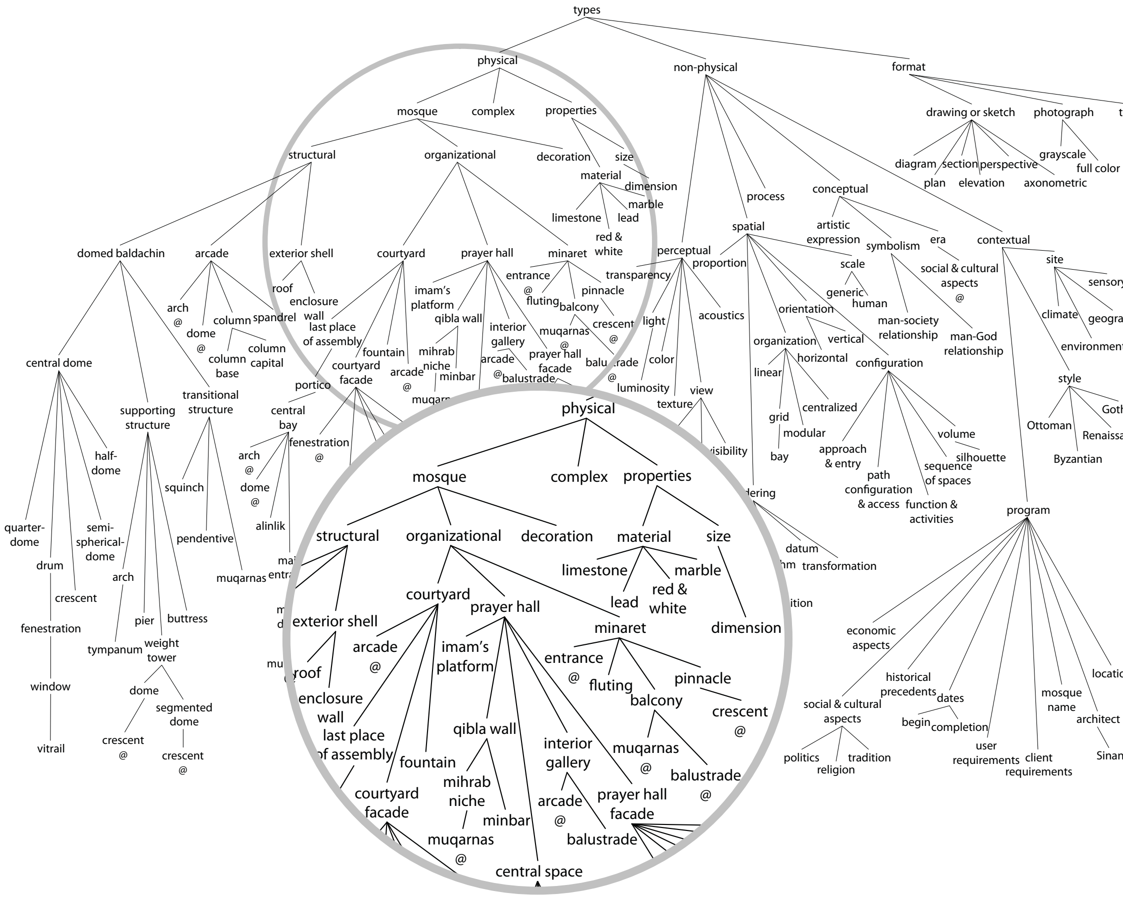


Figure 5.3. The concept network from the prototype application, defined for the categorization of abstractions of Ottoman mosques. The concepts that are defined elsewhere as substructures in the network are marked by '@'.

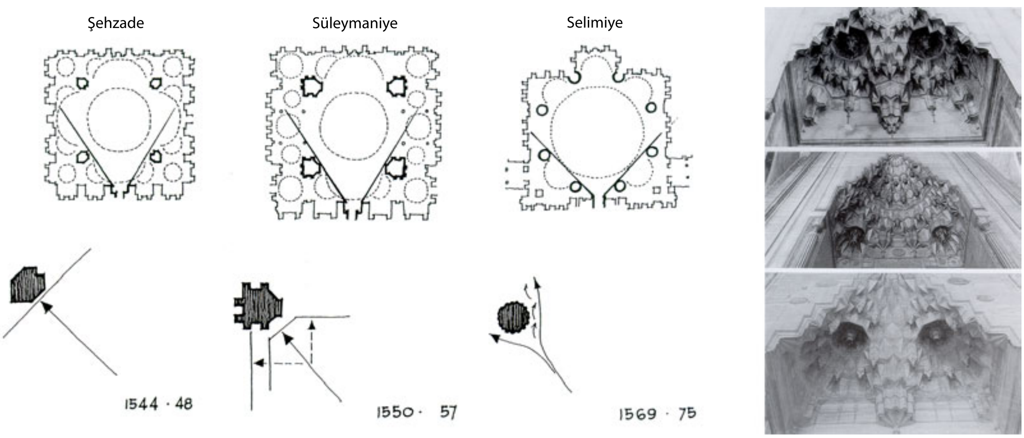


Figure 5.4. Three instances of types in classical period Ottoman mosques: left top, floor plans; left bottom, the role of structural piers on the field of vision; right, muqarnas above entrances from the respective mosques. Images from Egli (1997) and Erzen (1996).

around the definition of type and creativity issues in design. Types emerge from essential characteristics shared by a family of buildings (Figure 5.5). A type can be considered as a vehicle to formalize and reuse successful and reliable design solutions (Heylighen, 2000). The solutions embody valid principles that, as judged from examples, lead to adequate architectural qualities.

Types in architecture assist, besides the communication of shared knowledge, the analysis of existing buildings and the design of new buildings (Leupen et al., 1997). In analysis, one gives names to aspects of buildings and describes how these fit into a composition, resulting in an "analytical typology" (e.g., Flemming, 1990; Mitchell, 1990; Madrazo, 1995). In design, a reproducible system of design choices is stored in a "generative typology" (e.g., van Leusen, 1994; Achten, 1997; Gero, 1990). Within a generative typology, a type can be considered as bearing a specific design experience for a specific situation; a design aid. The typological studies of building morphology in historical analyses have established a rich body of architectural knowledge. These analyses define common categories of architectural form, such as volumetric organization, circulation patterns, axes, and boundaries (Ching, 1979).

Throughout history, there are two major approaches in looking at the concept of types (Aygen, 1998). The first one is an *a priori approach* which discusses that type is an extension of pre-existing categories (e.g., Abbé Laugier, Quatremère de Quincy, J.N.L. Durand) (Vidler, 1977). The second one is an *a posteriori approach* suggesting that types are defined by the comparison and grouping of existing architectural artifacts (Rossi, 1985; Argan, 1963). Moneo's (1978) definition of type is the most relevant to the goal of considering types as a formal structure for representing architectural knowledge:

> *"It [type] can most simply be defined as a concept which describes a group of objects characterized by some formal structure. It is neither a spatial diagram nor the average of serial list. It is fundamentally based on the possibility of grouping objects by certain inherent structural similarities. It might even be said that type is the act of thinking in groups."*

Moneo argues that architecture is not only described by types, but is also produced through them. The architect starts creating using types, later she can destroy it, transform it, scale it, overlap different types to produce new ones. She can use formal quotations of a known type in a different context as well as create new types. Moneo describes formal structure as:

> *"[...] centrality or linearity, clusters or grids, trying to characterize form in terms of a deeper geometry [...] reduces the idea of type as formal structure to simple abstract geometry. But type as formal structure is, in contrast, also intimately connected with reality, with a vast hierarchy of concerns running from social activity to building construction. Ultimately, the group defining a type must be rooted in this reality as well as in an abstract geometry. This means, for example, that buildings also have a precise position in history. ... This leads directly to the concept of a typological series that is generated by the relationship among the elements that define the whole. The type implies the presence of elements forming such a typological*

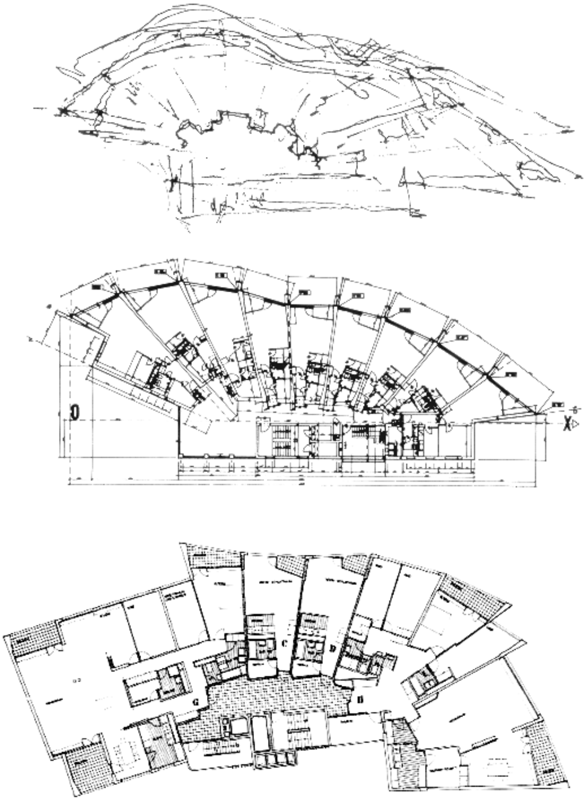


Figure 5.5. Variations on a plan type by Alvar Aalto. Top two images: Apartment building Neue Vahr, Bremen (Images from Reed, 1998); Bottom image: Apartment building Schönbühl, Luzern (Image from Jokinen and Maurer, 1998).

series and, of course, these elements can themselves be further examined and considered as single types; but their interaction defines a precise formal structure."

Using this approach to type, grouping physical or conceptual entities according to certain criteria provides the possibility to construct formal relationships between different architectural precedents. This approach can be used as a means in formalizing and classifying architectural languages according to different views. An architectural language is characterized by a vocabulary of elements and a grammar whose rules indicate how these elements can be placed in space (Flemming, 1990: 31). In this context, type implies the vocabulary of the architectural language and the underlying grammar. Types and typologies have been the subject of a number of computational systems where precedent knowledge is represented, usually in case-based systems (Pasman, 2003; Casakin and Dai, 2002; Aygen, 1998; Madrazo, 1999; Mubarak, 2004).

5.1.2. The representational structure of the application

The prototype application represents a complex information structure consisting of two structures: the network of concepts and relationships specifies the semantic structure, and the abstractions and their decomposition specify the document structure. XML has been used as the common syntax for the integral representation of these structures. XML is particularly suited to describe uniform and hierarchical data structures, especially in the form of images and text. The grammar of XML, i.e., the DTD, specifies the structure of both structures in the system: their elements, their nesting and additional properties, and their attributes (Figure 5.6). Both structures are recursively defined.

The semantic structure is defined in XML using the concept name as the tag, and by nesting the elements according to the network. Each concept is additionally identified by an ID, which is used for linking concepts to components. Below is a snippet of XML code for the definition of the semantic network:

```
<types>
  <typetree>
    <type id="t166">types</type>
    <typetree>
      <type id="t70">physical</type>
      …
    </typetree>
  </typetree>
</types>
```

Decompositions of abstractions are also expressed in XML. Documents are decomposed into their constituent entities and define the hierarchy of components. The abstractions in the form of images are broken up into sub-images by determining the important components, in correspondence to theconcepts, and by cutting them up using an image processing application (e.g., Figure 5.7). The abstractions in the form of text are immediately structured in XML. Each component is identified by an ID, and the component hierarchy is defined by using the ID as the index, and by nesting the elements. Concepts are assigned to components by their ID's. Below is a snippet of XML code for the decomposition of an image abstraction:

```
<document id="d6" types="t68 t66 t31" doctype="img"
content="sehzade17" title="plan and longitudinal section"
creation="2000-05-03 15:35:03" reference="3" width="769"
height="1075">
  <component id="d36" types="t68t t31t t66t" content="sehzade17-b"
  title="plan highlighting different zones" creation="2000-05-04
  12:49:06" width="769" height="489" xpos="0" ypos="494">
    <component id="d54" types="t48t" content="sehzade17-b-2"
    title="courtyard" creation="2000-05-08 10:00:42" width="423"
    height="489" xpos="15" ypos="494">
    </component>
  </component>
</document>
```

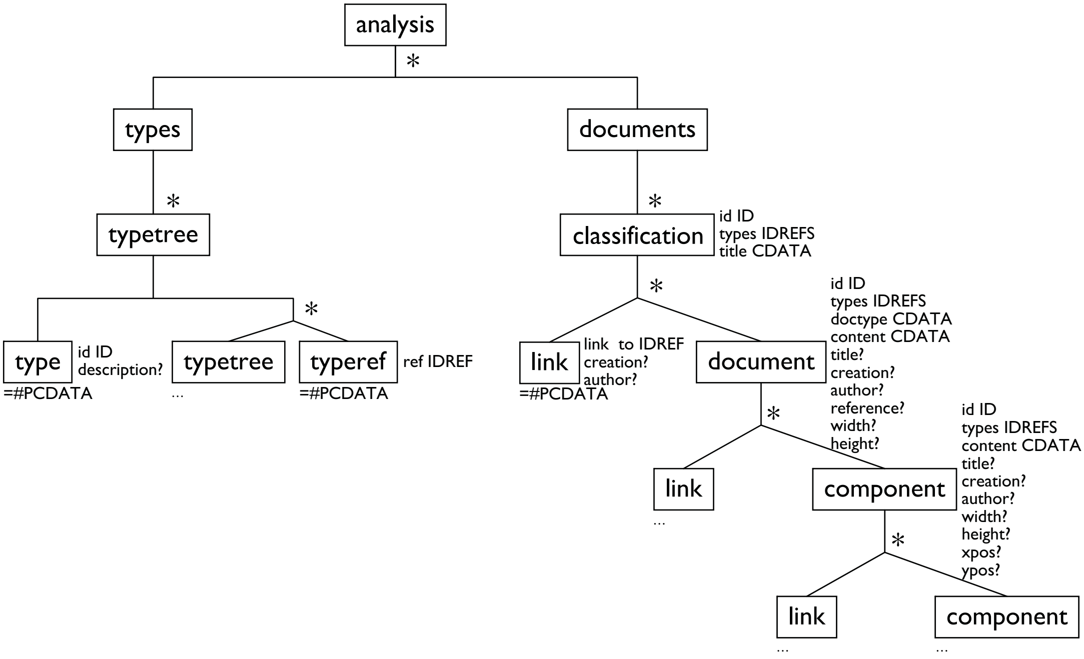


Figure 5.6. The recursively defined concept and document structures. The grammar of XML, i.e., the DTD, specifies the structure of both structures in the system: their elements, their nesting and additional properties, and their attributes.

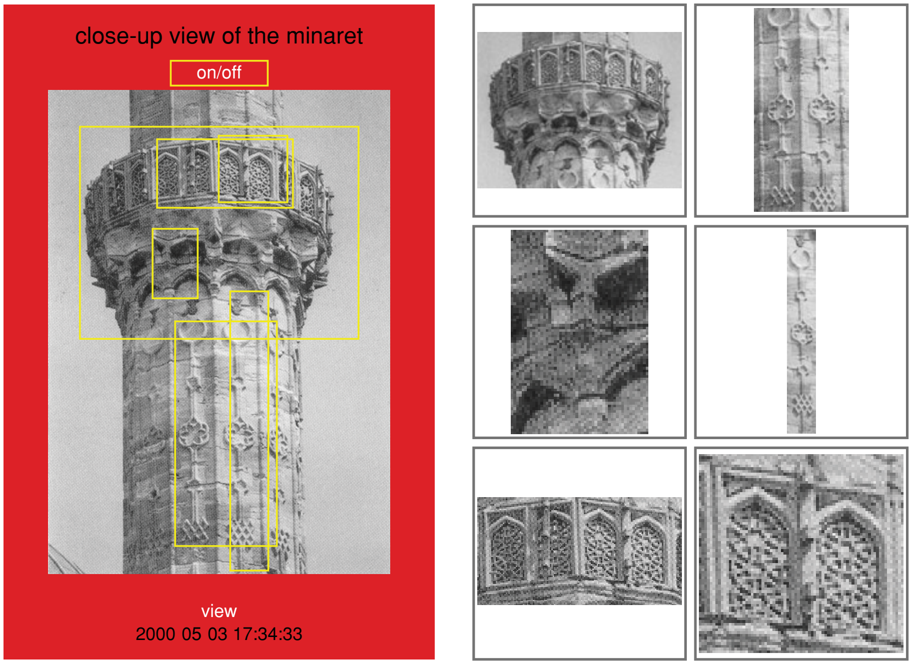


Figure 5.7. An example from the prototype application showing image decomposition.

In this organization, the document hierarchy initially relates components. However, these relationships are purely syntactical. Semantic relationships are added through the network of concepts and the assignment of these concepts to the components. Components that share the same concept are implicitly related. The concept network further relates components; these relationships are derived from the nesting in the concept network (see Figure 3.5). Finally, explicit relationships between components can be specified as references to the component ID's. These are transferred to the XML structure as IDREFS tags.

The resulting XML structure forms a flexible source for further manipulation and traversal. Components can be flexibly categorized and grouped according to their relationships and attributes, offering various views of the information structure. Views can be traversed and linked using both explicit and implicit relationships. The XML documents are transformed and visualized through related developments such as XSL, XSLT, Xpointer, and XLink.

In this application, for easy handling, the data structures are initially referenced and linked in the database, and later converted into XML structures. A MySQL database has been used for the storage of information, and a PHP script has been written to dynamically generate XML documents on demand.

5.1.3. The user interface and interaction of the application

The web-based interface allows the user to view both the semantic and document structures and their relationships. These views include both *in-world* and *out-world views* (Papanikolaou and Tunçer, 1999). An in-world view presents a component (or concept) together with its immediate neighbors within the structure, and displays all other components that share a concept with it (Figure 5.8a). The in-world view allows one to browse the structure and interpret relationships, and as such lets the user be guided to interesting out-world views. Out-world views offer an overview of (a part of) the information structure including all its relationships. These views provide visual feedback to the users on their traversals and offer selected detailed views by presenting the location of the currently viewed node within the structure. Out-world views also give an overview of the scope and depth of the semantic structure guiding the analysis. An out-world view developed in SVG is presented as a clickable map that offers an overview of the entire concept structure in relationship to the related documents (Figure 5.8b).

While the semantic structure itself contains valuable knowledge for a designer in the context of information gathering activities, it serves for the most part as a binding element in the structure providing relationships between the abstractions. When traversing the information structure, the content as available in these abstractions is the most important aspect. As such, while a component's concept, and its location in the concept network, may be presented as properties of the component, the relationships are specified primarily as component-to-component relationships. This not only ensures that links are presented as shortly as possible, facilitating a swift traversal, but also shifts the focus onto the content, rather than the structure that surrounds it. Concepts further serve a role as index to the information structure. Access to the analysis is provided through the collection of abstractions and from the concept network.

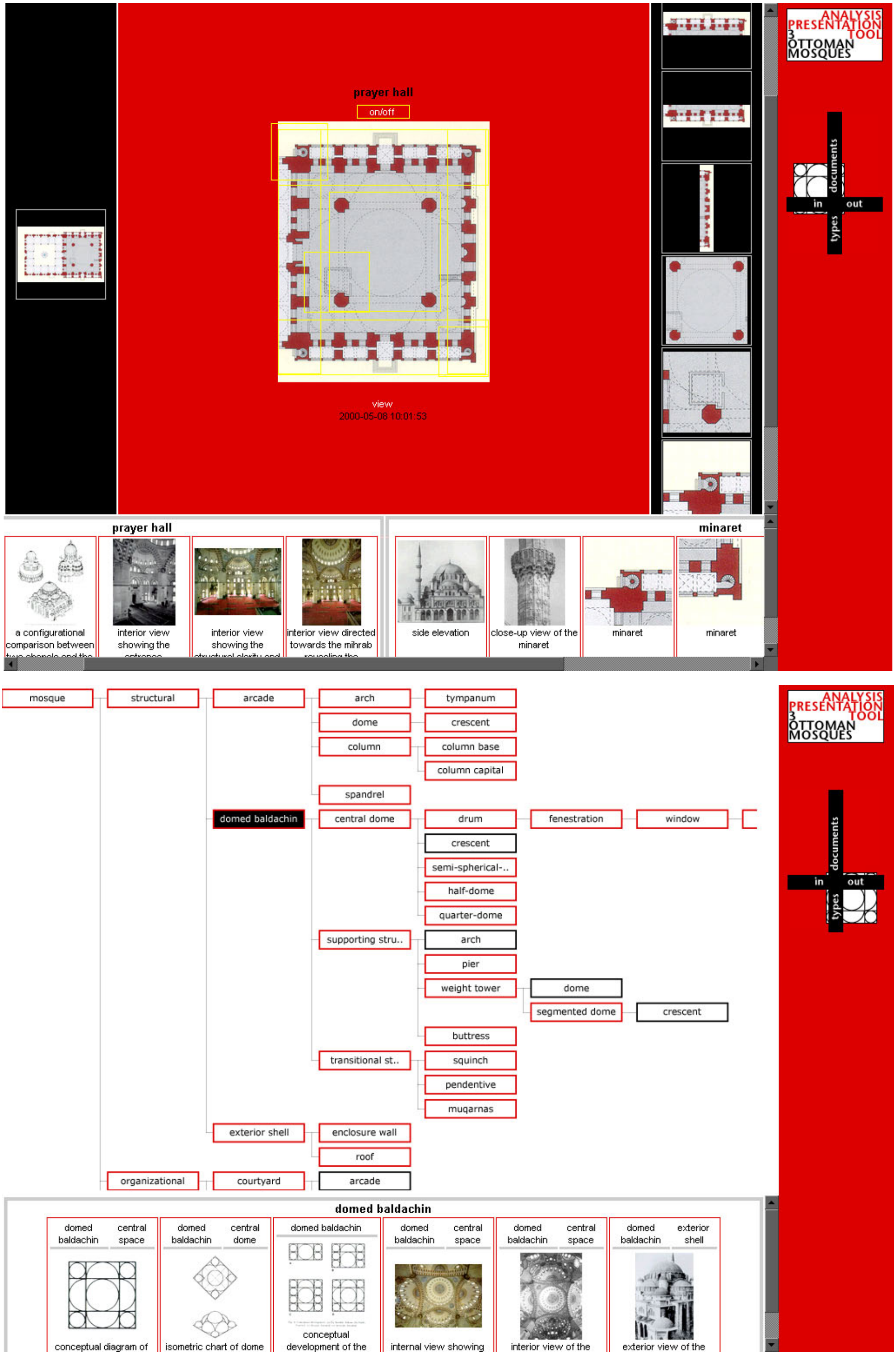


Figure 5.8. Two snapshots from the prototype implementation. a) above, in in-world view, b) below, an out-world view.

The presented approach offers the users a simple interface and easy mechanisms for the presentation and exploration of an analysis of design precedents from three Ottoman mosques, but it has no limitations on the nature of the content. The system is designed in a way that it can grow as users add documents from different buildings, even from their own designs. Since all the information is integrated within a single information structure, users will benefit from the different studies collected in the analysis, and can draw new conclusions across studies and presentations, potentially including their peers'.

5.1.4. Reflection on ArcIMap

This implementation shows the potentials of using ArcIMap in a specific context. Below a number of points are discussed that are derived from findings achieved by critically studying the use of the application. These points are meant as feedback for ArcIMap as a method and a model.

Integrated information representation: The information structure is represented in an integrated manner which makes the content in the application extensible and flexible. This application demonstrates that an integrated representational structure is successful in representing a complex information structure.

Associative browsing: The associative browsing mechanism implemented in the application seems promising. The way the information structure is conceived and presented allows and encourages the designer to use it for various purposes: for inspiration, for looking up specific knowledge, and for following relationships and building up a cognitive path that enhances the design thinking process. The associative browsing mechanism implemented in the application allows users to formulate planned and opportunistic browsing strategies using cognitive and perceptual determinants (see Section 2.4.4). The out-world view designed as an indented list, however, is not quite intuitive to use. A more user-friendly view for the semantic network structure is needed. Figure 5.9 presents some exemplar out-world views that were developed as clickable maps that offer an overview of the entire concept network in relationship to the related documents. Such intuitive views have been used in further implementations of ArcIMap and are described in the following sections.

Time and effort needed for inputting information: The information in this application has been supplied and input by one person. However, the preparation of documents and document decompositions, the semantic structure, and the definition of associations among documents and elements of the semantic structure cost a considerable amount of time. Depending on the intended use context, i.e., educational, in a research setting, or for use in practice, appropriate mechanisms and processes need to be in place in order to enable an efficient, accurate, and easy to use input of information into the application.

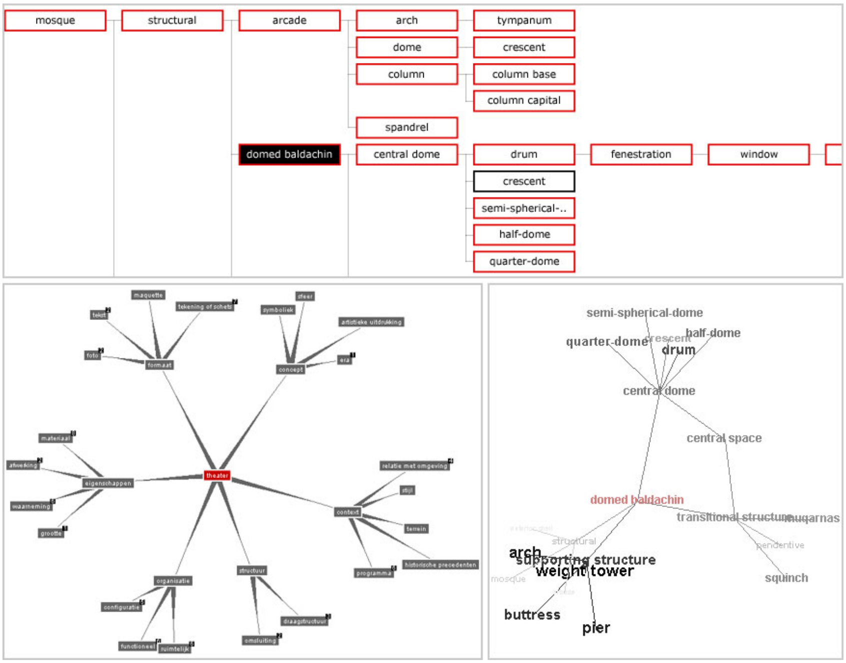


Figure 5.9. Three snapshots from out-world views of the semantic network of concepts. a) a 2D list view, b) a 2D dynamic tree view, c) a 3D dynamic network view. The focus of this figure is on the graphical representation of the structures, not on the concepts themselves.

5.2. BLOB INVENTORY PROJECT (BLIP): AN INTERACTION STUDY OF ARCIMAP

Blob Inventory Project (BLIP) is a web-based environment[45] for modeling knowledge[46] that has emerged from digital design, engineering and production processes of free form geometry buildings[47] (Kocaturk et al., 2003; Kocatürk and Tunçer, 2004; Tunçer, 2007). The premise of this application is that with the realization of such buildings, new processes have emerged that cross the boundaries of the working processes of architects, engineers and manufacturers (Chaszar, 2006). Since decisions taken in one domain immensely influence processes within other domains, an integrated knowledge of interdisciplinary processes is essential especially in the early stages of design (Sariyildiz et al., 2002). In this context, BLIP offers architecture and structural engineering students, researchers, and practitioners an expandable knowledge base of built and unbuilt examples of double curved surface buildings organized around formal, structural, and production processes. Architecture students develop fancy double-curved surface architectures in their design studio projects, but they generally do not have much knowledge about the constructability of their design, choice of materials, and structural and manufacturing implications of these forms. BLIP can greatly extend their knowledge about constructability issues, and more importantly, set them to think about such issues.

BLIP represents and displays a complex information structure. In BLIP, the semantic structure is initially defined by a domain expert and is not freely adaptable by the users of the system. This structure is defined as a hierarchy of context independent concepts of design, engineering and production processes related to free-form buildings, organized in three main branches corresponding to domains of *formal aspects*, *structural aspects*, and *production aspects*. Later, concepts below the three branches of the hierarchy are interrelated, making the structure into a network. These interrelationships represent the emerging relationships between formal, structural, and manufacturing aspects of double-curved building processes. The document structure consists of precedent structured web pages representing precedent information. These are associated with concepts and conceptual relationships in the semantic structure. Figure 5.10 illustrates the overall information structure underlying BLIP. The semantic structure serves as the main access point into the information structure. Users input documents and new interrelationships into the application through a separate 'management interface'.

From the viewpoint of implementing specific aspects of ArcIMap, in addition to the underlying complex information structure and the technical infrastructure as a tool, BLIP focuses on the cognitive and process related aspects of creating interrelationships between concepts, as well as the user interaction of the application. The definition of these interrelationships goes hand in hand with the inputting of precedents in the application, because the knowledge acquired by analyzing precedents reveals new process relationships. The process of defining an interrelationship requires users to browse the

---

[45] The system has been implemented based on the InfoBase database and its functional library (seeAppendix B).

[46] BLIP has been developed as a joint work by three PhD researchers (Bige Tunçer, Tuba Kocatürk, Martijn Veltkamp). Tunçer's research provided a flexible and extensible framework for knowledge modeling that acts as the backbone of the information and interaction structure of BLIP. Kocatürk's and Veltkamp's research provided the main context and the related knowledge content for the application which contributed to the cross-disciplinary richness of the knowledge content due to the separate research foci and disciplinary background of the two researchers. Joost Beintema contributed to the programming of BLIP.

[47] Blob is a word used for free form geometry buildings.

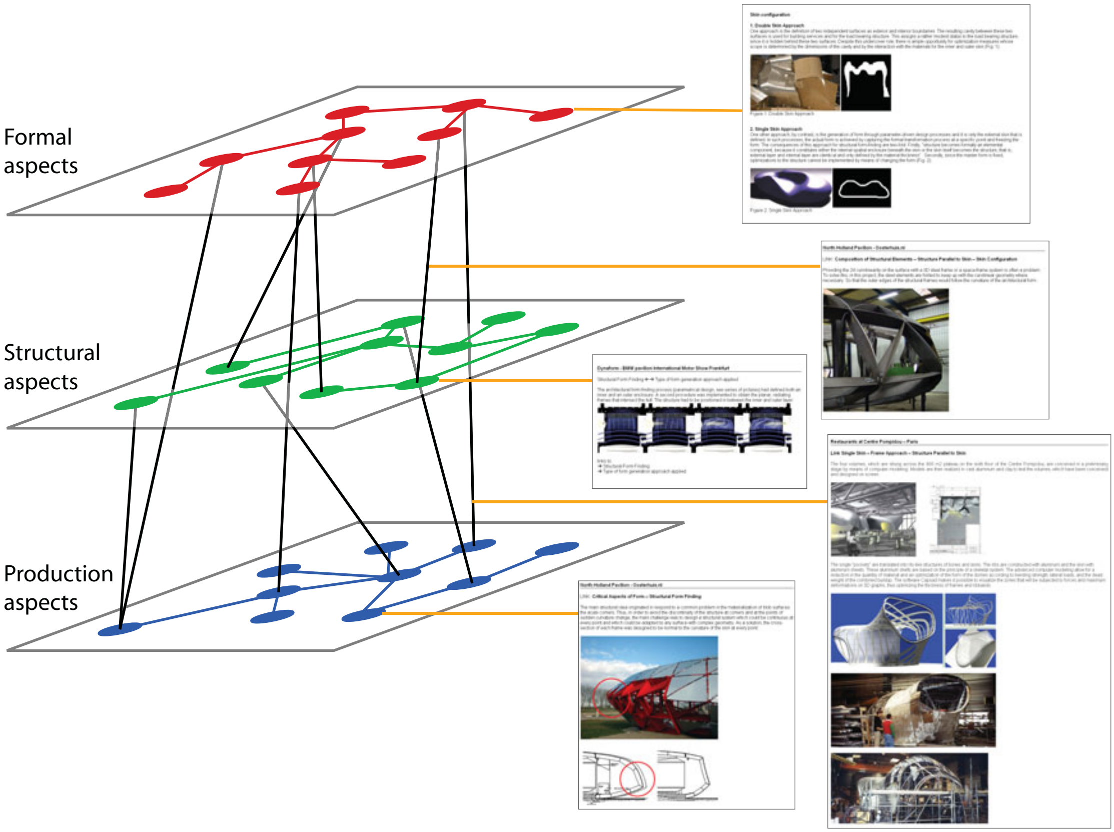


Figure 5.10. The information structure of BLIP. Concepts are structured in three levels: formal, structural, and production domains. Concepts in each level are related through hierarchical relationships. Additional relationships can relate concepts from different levels. Documents are related either to concepts from any level, or to cross-domain relationships.

existing semantic structure and document structure, select two or more concepts to relate, and input a new relationship and a new document. BLIP is not meant to entirely support the activities of a community of practice, therefore it cannot be described as a complex adaptive system. The 'correctness' of the information BLIP contains is important, and a moderator checks for this.

The BLIP interface displays the predefined concept hierarchy on the left hand side (Figure 5.11). The three main branches are displayed in a collapsible list, the selected concept is highlighted. The dynamic network visualization view of the same hierarchy and the interrelationships is displayed in the top left area[48]. It is a dynamic view with options for zooming and displaying levels of detail. Figure 5.12 displays various views of the semantic structure in various levels of detail and zoom degrees. This view is synchronized with the

[48] This is a Java application developed on top of the open source TouchGraph application (http://www.touchgraph.com/).

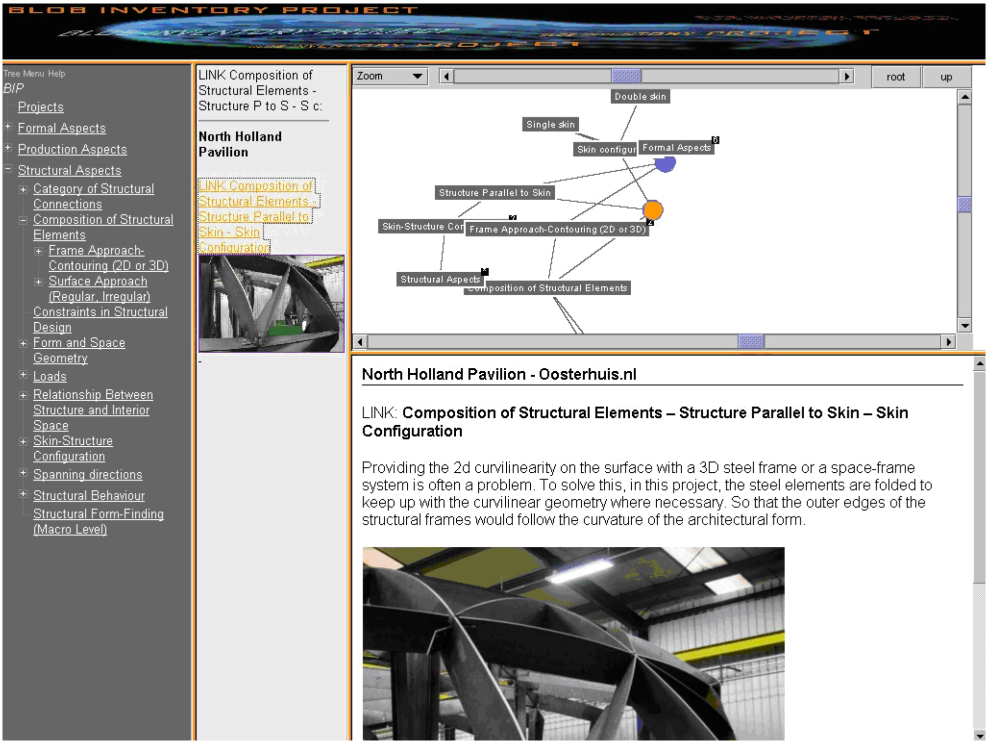


Figure 5.11. A Screen-shot of the interface; Frames: Left) predefined concept hierarchy, Top left) dynamic network view of the semantic structure , C) thumbnails of related documents, D) content of selected document. Clicking on an inter-relationship between concepts brings up the related document that describes the meaning and nature of this relationship contained in a document.

collapsible hierarchy view. Selecting a concept or an interrelationship in either of these views displays thumbnails of the related documents that contain information on the selected concept in the area next to the concept tree view. When a thumbnail is selected the document content appears in the bottom left area together with its associated concepts and relationships.

BLIP has been used in an educational context in an elective M.Sc. course[49]. The students of this course used BLIP to store, retrieve, and exchange information that they generated by systematically exploring multi-disciplinary digital design solutions according to the given set of domain related constraints.

[49] This M.Sc. third semester course was a Technical Study in the Emotive Architecture program (BKM3AUE2), at the Faculty of Architecture, Delft University of Technology, taught by Tuba Kocaturk and Martijn Veltkamp.

5.2.1. Reflection on ArcIMap

The use of BLIP in education has been studied and evaluated in order to generate observations and feedback for ArcIMap as a method and a model. The evaluation has been conducted as a semi-structured two hour interview held with the main instructor of the course and observations of the author during the course. Some important evaluation results related to the reflection on ArcIMap are listed below.

Semantic relationships: When expressing inter-relationships between concepts, the nature (meaning) and the directionality of the relationship turned out to be crucial. When a student created a new conceptual relationship in the system and input a document clarifying the relationship, other students did not easily understand the nature of the relationship, i.e., why it was there, and also if the relationship had an origin, i.e., if the relationship was from one concept to the other, or bi-directional. This observation reinforces the need for explicit semantic relationships in a semantic structure.

Partially predefined semantic structure as a cognitive aid: *It was observed that students took the predefined concepts as the basis of their analysis of precedents.* Having a predefined conceptual structure made the analysis process easier for them. However, having a predefined concept hierarchy also dictated a design and analysis method to students, although this was never explicitly the stated. On the other hand, since students in general do not have a lot of design and analysis experience and conceptual thinking is not very easy for them, this way of working somewhat simulated expert behavior for them. Therefore, the designer(s) of an educational application of ArcIMap must carefully consider if and how much of a guidance they want to provide to the users by defining a portion of the semantic structure in advance. This depends on the specific context and the knowledge and experience level of the intended users.

Interaction: The use interface and interaction worked considerably well for the intended context and users. Users particularly found the dynamic view of the semantic structure view easy to use, intuitive, useful, attractive, and fun to use. However, it was observed that since the semantic structure was rather large, it was challenging for the users to find where they were previously in the browsing process if they wanted to return to a concept they earlier saw. Functionality for showing a browsing history would help with that.

Time and effort needed for inputting information: It seems that for educational applications in general the time and effort needed to input information into the application is not an important issue if there are no technical problems. There is no obvious time and financial pressure on students when using such applications. The task of information supply is usually incorporated in the didactic approach as assignments for students to prepare documents, think about concepts and conceptual relationships, etc., so the students are expected to spend some time on this.

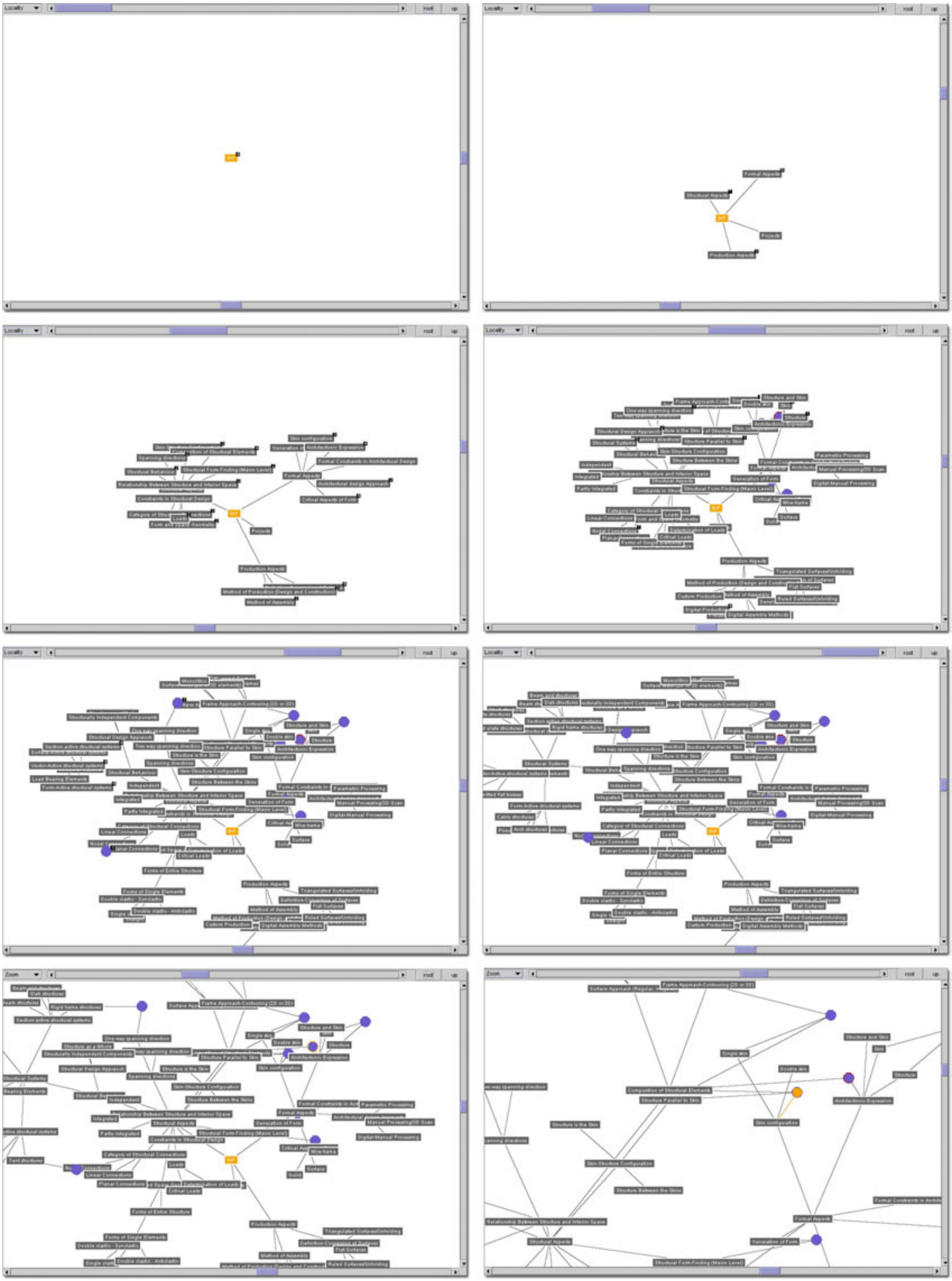


Figure 5.12. Various views of the semantic structure in various levels of detail and zoom degrees. Users of the application explore this structure to familiarize themselves with the knowledge structure. Users also use this view to identify locations for the definition of new conceptual relationships.

5.3. DESIGN ANALYSIS NETWORK (DAN): AN EDUCATIONAL STUDY OF ARCIMAP

Design Analysis Network (DAN) is a prototype architectural analysis application that implements ArcIMap and demonstrates a procedural approach of how to work with ArcIMap in an educational environment. DAN is a web-based environment[50] for the construction and presentation of a body of architectural analyses in the context of a design studio. In architectural education, as in architectural history, theory, and design, complete and thorough analyses of architectural bodies and objects are indispensable. These analyses cover many different aspects of the subject, e.g., physical and contextual attributes as well as geometric, functional, typological and organizational relations. When the results are computationally integrated, new views and arguments can be deduced from these that transcend the individual analyses. DAN creates an extensible and cooperative library of architectural design analyses, searchable by content, and instructional for coming generations of students. The result of the analyses is collected in a common library such that students, in later design activities, can draw upon other students' results for comparisons and relationships between different aspects or buildings. DAN also acts as a digital presentation environment for students, where they can present their analyses to their design studio instructors.

5.3.1. The design studio context

DAN has been used in two iterations of instruction in the fourth semester (second year) design studio at the Faculty of Architecture, Delft University of Technology by 194 students. The central design theme of the fourth semester design studio is a 'small public building'. The first time DAN was used the project was a small theater and, the second time a small museum. In the studio, the students are given a relatively complex functional program and are requested to design and work out the materialization of this building. The students begin the studio by analyzing selected precedents (historical and contemporary) of the relevant building type with respect to various criteria (composition, program, construction, context, type, etc.) and from structural, formal, and functional points of view (Figure 5.13). Documentation of these precedents is presented to the students in the form of drawings, pictures, and texts. Until the use of DAN, such documentation was solely provided in the form of a book. DAN has a precedent library component[51] that makes this documentation available on the web (Figure 5.14). Besides the material found in the precedent library of DAN, students are also encouraged to do web searches to find additional relevant information and store the relevant information in the DAN precedent library.

Traditionally, in an architectural analysis, students build up declarative and procedural knowledge about their profession. In the design studio education, students gather documentation about a specific building type, they investigate the goals of the architect, the instruments the architect uses, and they do a critique of how these instruments have succeeded to reach the original goal. Experienced designers don't need to perform structured analyses to gain this knowledge. They already have the cognitive constructs to

[50] The DAN environment uses the InfoBase database and part of the InfoBase functional library. More information about the InfoBase project and its results can be found at http://infobase.bk.tudelft.nl. A schematic overview of the InfoBase database can be found in Appendix B.

[51] The precedent library component of DAN has been partially developed by Rubiën Grootfaam.

Figure 5.13. A design analysis critique in the 4th semester design studio. Students present their analyses to the design instructors in a studio critique setup.

be able to understand or visually 'recognize' how a building 'works'. This knowledge is internalized as procedural, even tacit knowledge. Novice designers, however, must learn by doing formal analyses to gain this knowledge. However, there is a danger in this formal analyses process that students tend to draw mindlessly colored areas on plans, claiming that these are the result of an analysis. In reality, these are the result of some mechanized analysis efforts, and the students do not really gain the procedural knowledge of the physical and conceptual schemata in the design precedents. Students need to be aware of and think about every action they perform during an analysis, and why. Therefore, they need to explicitly make claims on each step of the analysis, and also, provide explanations to graphic notation they are using during the analysis (Steenbergen et al., 2002). Otherwise, the analysis may not exceed a set of colored areas. DAN enforces students to make claims on the rationale of the analysis and also on the content of the analysis. This is a good start for the students to build up knowledge, because they need to think about the analysis process in a structured manner.

The use of DAN in the design studio was integrated in the design project[52], but was officially part of an Informatics course, which is a small course with few credits. This is important information because of the way students perceived the weight of the assignment. The DAN assignment was completed in a workshop of 3.5 hours taking place in a computer lab environment each time with around 60 students. This workshop was designed in communication with the design studio coordinator, who was also an instructor. The content of DAN reflected on the official semester book. The workshop started with an explanation session of about half an hour. Then the students completed the assignment, asking questions to the teachers if necessary. Students teamed up in groups of two to complete this assignment.

The assignment given to the students was to do two analyses of a selected building from the DAN precedent library (Figure 5.14) according to two criteria, to write a short concept document for their own design, and then to do a search in the database in order to find

[52] The design instructors of this studio were Herman Prast, Herbert van Hoogdalem, and Steven Steenbruggen.

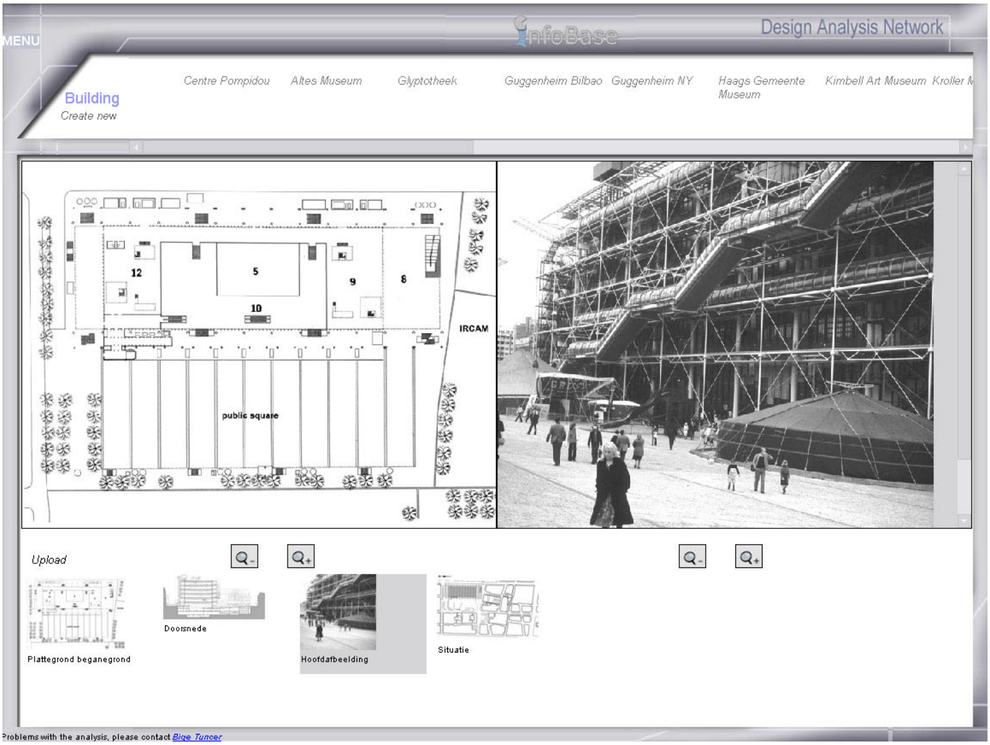


Figure 5.14. Interface of the precedent library component of DAN. This component acts also as the starting point of an analysis.

analysis examples related to their own design concept. Appendix A presents the assignment and the description of the environment as given to the students.

5.3.2. The semantic structure of the DAN information structure

In the context of creating an environment where students learn to express meaningful choices when doing an architectural analysis, and applied to the context of architectural education, educators want the student or future architect to learn to lay claims on data collections (see Section 2.3). Therefore, in the context of this design studio, the educators teach students to handle and use metadata along four quality dimensions: these represent constructive, relational, objective and subjective qualities (Groen et al., 1980; Kooistra, 2002) (Figure 5.15). Constructive quality (the concept) signifies the *will to design* (the will to improve) that arises from the necessity to achieve agreement through correspondence. It is an intuition that springs from comparison. However, constructive quality has no meaning unless one is able to persuade other people (architects/designers) to believe and invest in one's construction. Thus, objective quality (the matter) is also needed; one needs repeatable observations, scientific facts. Unfortunately, facts don't have a meaning without human intentions, wishes, or drives. Therefore, intention represents the third kind of quality: relational quality (the user). Producing survival knowledge relates

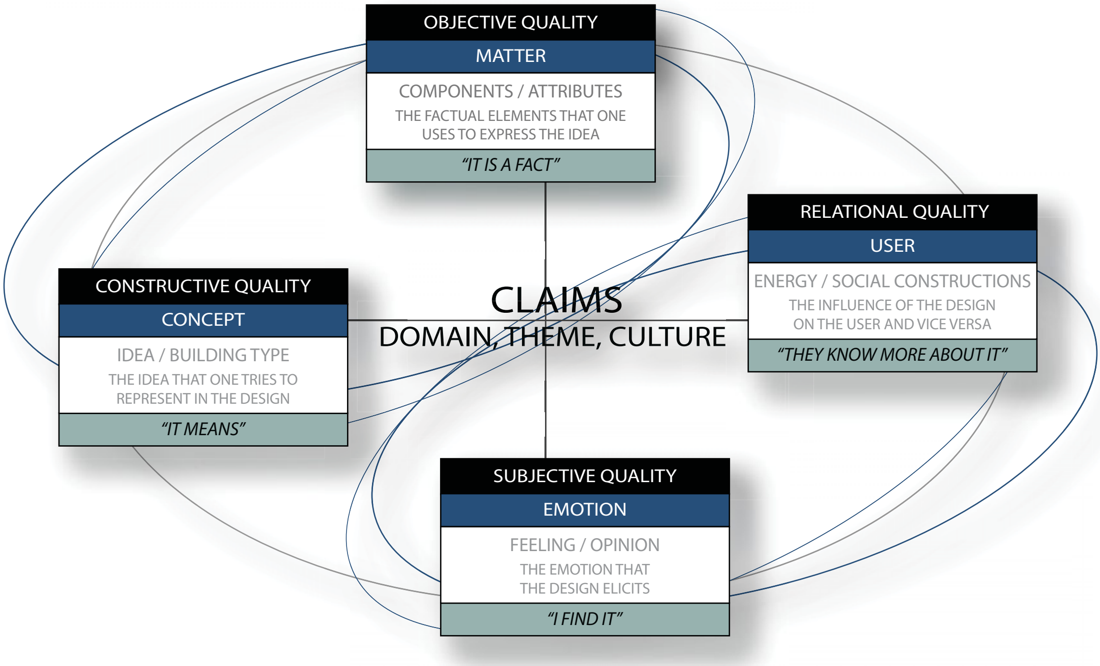


Figure 5.15. Diagram denoting the space formed by four dimensions of quality claims and their interpretation for architecture.

people; survival is a basic activity not unknown to students in an educational setting. Lastly, the fourth kind of quality is the subjective quality (the emotion). This kind of quality signifies one's personal taste, conviction and interest. Expressing this quality is presenting what one is thinking and feeling and through that it represents one's position in the quality system. Special about subjective quality is that it can exist only in *opposition* to objective quality. Objective quality is a (scientific) domain. Subjective quality simply disappears if it comes within the range of this domain.

The semantic structure of DAN has been organized around these four qualities[53]. The analysis is carried out according to the concepts described in the four dimensions of claims. Table 5.1 lists the claims in the relational, constructive and objective dimensions that were provided to the students by the design instructors. Students can suggest new keywords to an instructor. The relational claims represent the main guiding principles of buildings. The constructive claims represent the instruments used by the architect to

[53] As an example of how the four dimensions of quality claims work, the italicized terms can be considered as keywords or claims. As a general example, an *art program* on TV is a construct that considers *art loving viewers* (relations). The program presents *facts about theatre shows* that may convince the viewer that it is *worthwhile to attend a theatre show* (emotion) – upon which the viewer may send the program a *disappointing* (emotion) review. Considering the field of architecture, the *theatre* is a construct that considers an *audience* looking for *entertainment* (relations). The theatre has a (large) *hall with 1500 seats* (fact). The audience ascertains that the *seating is comfortable* (emotions).

Table 5.1. A list of the concepts included in three branches of the concept structure: relational, constructive, and objective claims. These were presented to the students in Dutch, they have been translated to English here.

Relations (relational claims)	**Constructs (constructive claims)**	**Facts (objective claims)**
Constructional organization	Finishing	Elevation
Flexibility in use	Structural system	Axonometric view
Functional organization	Utility systems	Diagram
Light as design instrument	Color	Section
Sizing systematic	Massing	Photograph
Relation to existing buildings	Material	Scale model
Relation to existing infrastructure	Enclosure system – roof	View of the surroundings
Relation to green areas	Enclosure system – building envelope	Perspective view
Relation to surroundings	Organizational pattern – centralized	Floor plan
Routing	Organizational pattern – clustered	Sketch
Spatial organization	Organizational pattern – radial	Site plan
	Organizational pattern – grid	Text
	Organizational pattern – linear	
	Proportion	
	Rhythm	
	Functional unit – administrative space	
	Functional unit – auditorium	
	Functional unit – circulation space	
	Functional unit – entrance	
	Functional unit – restaurant, cafe	
	Functional unit – car parking	
	Functional unit – service space	
	Functional unit – exhibition space	
	Scale	
	Symmetry	
	Texture	
	Transparency	
	Construction process	
	Circulation system	

Table 5.2. A selection from the list of the subjective claims included in one branch of the concept hierarchy. These were freely defined by students. They were originally in Dutch, they have been translated to English here.

Emotions (subjective claims)		
Potato Deviating form Separate cubes form an entity Image forming Boring entrance block Peculiar form of entrance A lot of light and windows Complex Contrast hi-tech with green Closed Little character Burst through form Section is not clear Box Really gives an amphitheater feeling Clear Clear where the entrance is Clear functional separation Clear relationship between texture and water Simplicity Smooth Light Simple Organic Effective multi-use of space Back to the 70's Building as a bridge Building intertwined with surroundings Building shows clearly that it's a museum Closed character Structured Cosy Nice skylight Green Big gesture in context A lot of transparency Heavenly	Too much repetition Entrance as temple in abbey garden Complicated Classical Small Small café for large museum Colorful Layers Long Empty spaces Ugly Ugly square with parking Nice square with terraces Light and space Light and simplicity Light, peaceful hall Line Beautiful Beautiful interior Beautiful use of color Beautiful color variation through light reflection Beautiful height difference Nice interaction of forms Nice flowing wall Nice light-shadow interaction Nice transition between the old and new parts Neutral white façade Not worked out, but good start Open Open character Open atmosphere Striking Striking icon in the surroundings Striking use of materials Orderly Well organized Half finished	Practical Rational composition Spacious Space in space Insightful Change of speed towards city and park Spiral Spiral movement downwards Stupid Austere Austere glass façade, doesn't fit with the surroundings SUPER BEAUTIFUL Traditional Elaborate A lot of important monuments in the neighborhood Far away Hidden Avoiding museum fatigue Refreshing Contaminated Boxy Form originates from location Form originates from function Strange Entrance doesn't attract much attention Ziggurat form evokes harshness Floating Crooked ramp, not good for disabled people Flat Postmodern Fantastic Calming Peaceful inner courtyard Crooked Intense

achieve the desired result. The objective claims represent the type of the design document. These objective claims are mainly used for describing the representational medium that is used to express the ideas and, furthermore, to assist in searching and retrieving from the analysis library later during the assignment. Students, however, freely add keywords to the list of subjective quality claims. This keyword expresses their opinion or subjective view about this aspect of the design. Table 5.2 presents a selection from the collective set of subjective claims created by the students.

When students select keywords from all four dimensions, this set of four keywords forms a unique "*key*" (Stouffs et al., 2004b). *KeySet* provides each work with a key consisting of a combination of four or more keywords. These four dimensions correspond to the four dimensions of socio-cultural qualities presented above and define the space in which the design process takes place. The students are warned that associating keywords or claims to their work is relative—it always involves interpretations. Thus, the key (as a collection of keywords) a student assigns must be communicable. Furthermore, considering that every design or analysis is unique, the students are taught that chaos arises naturally when 'populating' a database or library. Every analysis that is submitted 'queries' the DAN analysis library and forces it to position the analysis. Since every analysis is unique, it receives a position that does not coincide with any other. In this way, the DAN analysis library ends in 'chaos,' unless a constraint is imposed that applies with the input of data. This constraint is imposed with the aim that students are encouraged to learn from one another and work together, at their own initiative. Specifically, human communication constraints the uniqueness of the designs as positioned in the DAN analysis library through the use of metadata. This introduces the principle of order (and, with it, that of simplification) and self-organization; "order arises from complexity through the process of self-organization" (Prigogine and Stengers, 1984). The obligation that correspondence on claims must take place in the DAN analysis library can be regarded as self-organization in this respect. The DAN analysis library can be considered a self-organizing system exactly because the content is placed under the condition of human concepts that can be exchanged through correspondence.

5.3.3. An architectural analysis process in DAN

The first step in the analysis is the selection of a building to analyze from the collection presented in the precedent library (Figure 5.14). This library offers an overview of documented precedents, organized according to the name of the building. The process of documentation forms an integral part of the analysis process in this design studio. Subsequently, the students (in groups of two) choose one of the visual documents belonging to the selected building (most of the time a plan or section) as basis for their analysis. They load this selected document into the *DAN toolkit* to carry out of the analysis.

The DAN toolkit contains a number of tools: to draw markers (e.g., sections and views) and color coded areas on a document (e.g., a plan or section), annotate these, relate these to other documents, link them to appropriate concepts within the semantic structure (Figure 5.16a), and then generate web pages from these, as entry pages to analyses (Figure 5.16b). The DAN toolkit has been implemented using Java[54].

[54] The Java implementation has been done in part by Henry Kiksen.

In the analysis of the selected precedent building, the student investigates one of various criteria, for instance, composition, program, construction or context. The analysis process starts with the students selecting an aspect that describes the purpose of the analysis. In this way, students select which aspect of the building they wish to analyze. This decision comes from studying the guiding principle or principles that played a role in the design of the selected building. For example, the movement route of the visitors has played an important role in the design of the Kunsthal in Rotterdam, designed by Rem Koolhaas. Students select a concept from the list under the relational claims for this purpose. For example, choosing 'routing' for Kunsthal would be a good choice. When the students select the entire document that is loaded in the toolkit, this selection is shown as a highlighted (red) rectangle enclosing the plan (Figure 5.17a). This being selected, students then select at least one concept from the relational, constructive, and objective dimensions. For example, a constructive and an objective claim for the Kunsthal – Rotterdam could be respectively "circulation system" and "floor plan". They also type in a keyword as a subjective claim – for example, "genius".

Next in the analysis process, students start working out the analysis. They need to demonstrate the content of the analysis. This is usually done by drawing diagram-like notations on a plan, for example to show the circulation scheme, or the functional zoning, or the level differences on the section, etc. Using the DAN toolkit, students can draw a number of markers to express these diagram-like notations on the base document. The toolkit offers the user the ability to draw colored areas, section markers and view markers on a plan, section or elevation (see Figure 5.16 for the various markers integrated in the toolkit). Students can select a color for these markers. Each marker must have at least one concept from each semantic structure dimension associated with it. In addition to selecting a keyset for each marker, students also associate other documents that reside in the repository to this marker (Figure 5.17b). These documents are automatically loaded into the toolkit when starting the toolkit. Associating a document to a marker is done by selecting the marker and clicking on one or more document thumbnails on the right hand side of the interface. Considering that the result of this analysis will be presented as a clickable image map with hyperlinks to all associated documents, in this way, one can create a sequence of images, a digital story, marking sections on images, linking them to other images, following a story. Since the product of the DAN toolkit is an image map, one can simply click through a sequence of images with claims, following scenes of a storyboard, telling an architectural story.

In addition to the mandatory association of concepts and the possibility of associating other documents to markers, it is also mandatory to add a short annotation to each marker. This annotation is meant to clarify the purpose and the meaning of the student's action, in order to allow for a self-explanatory analysis. In addition to the concepts, the annotations enable the transfer of knowledge to other students, and instructors, who retrieve the analysis later from the repository of analyses.

The result of the analysis is presented as an image map (Figure 5.16). Such an image map can serve as an entry page to the analysis, or as a content map or index to a collection of related documents. When one moves the mouse pointer over a marker, a preview of the associated claims, the related documents, and the annotation appears. Markers can be clicked on to browse to the respective documents. The examples in Figure 5.18 belong to the 'print view'. This view shows the analysis result in a way that can be presented to the design instructors. It numbers the markers, and shows all the associated claims, to the

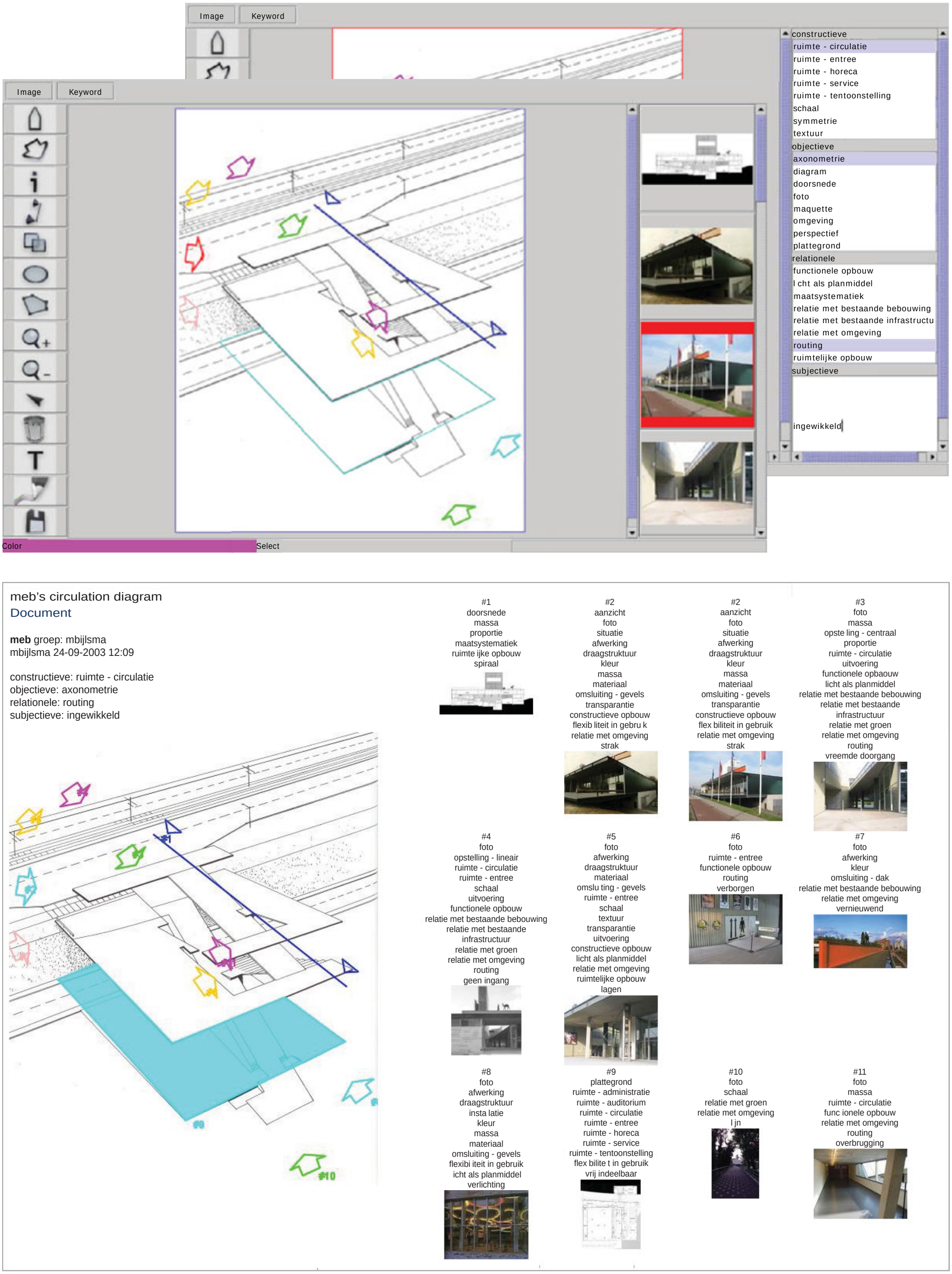
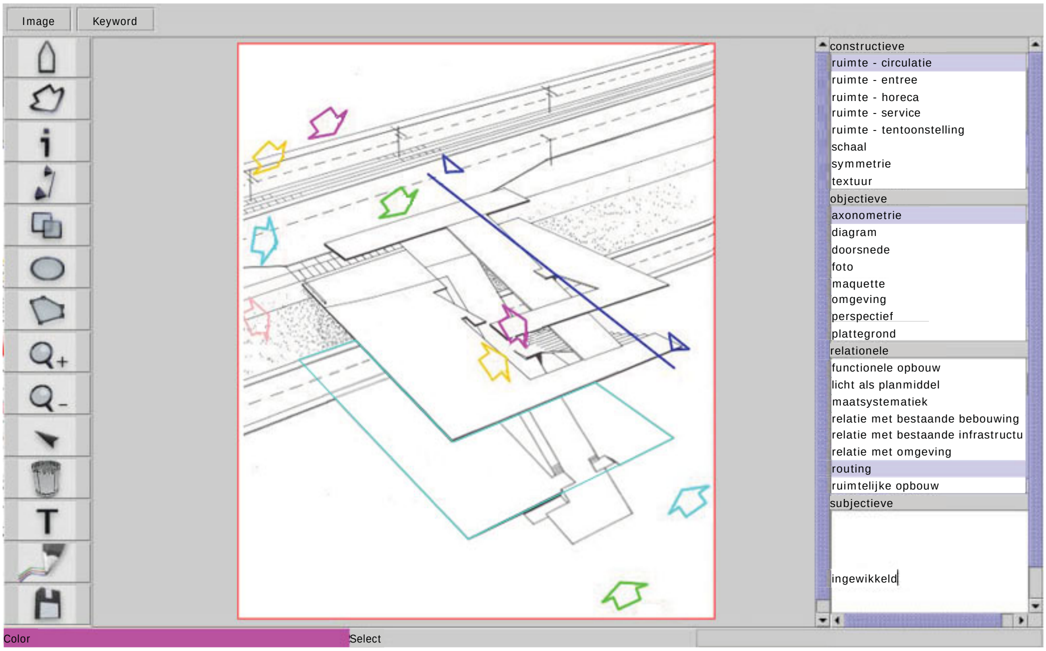


Figure 5.16. An analysis carried out in the DAN toolkit and the resulting image map. a) Above, snapshot of the application that serves to generate image maps using various markers that are related to other documents or keywords. b) Below, resulting image map with section, elevation and view (photo) markers.



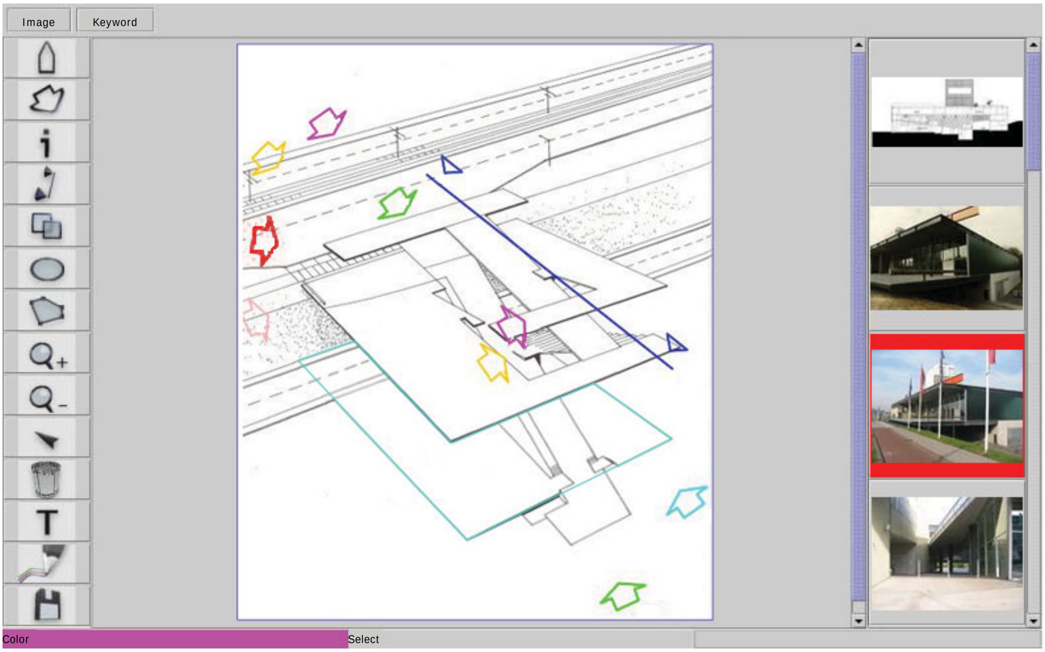
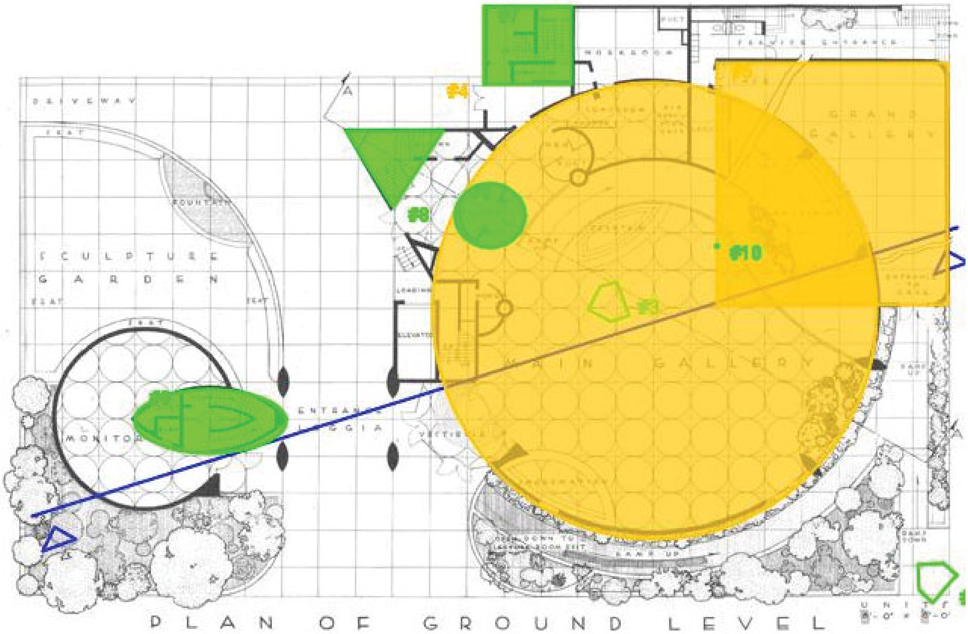


Figure 5.17. The interface of the DAN toolkit. a) Top: Markers can be related to claims (right); b) Bottom: Markers can be related to other documents (right).

constructieve kwaliteiten: draagstruktuur
objectieve kwaliteiten: plattegrond
relationele kwaliteiten: functionele opbouw
relationele kwaliteiten: maatsystematiek
subjectieve kwaliteiten: vorm compositie



#1
doorsnede
proportie
ruimtelijke opbouw
verduidelijking opbouw

#2
foto
massa
ruimtelijke opbouw
beeld bepalend

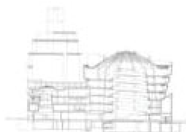
#3
foto
ruimte - tentoonstelling
licht als planmiddel
ruimtelijke opbouw
mooi

#4
foto
ruimte - tentoonstelling
functionele opbouw
vorm bepalend

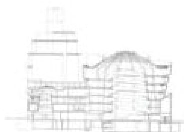
#5
foto
massa
ruimtelijke opbouw
afwijkende vorm

#6
foto
ruimte - circulatie
routing
vorm herhaling

#7
doorsnede
ruimte - service
functionele opbouw
vorm doorbrekend



#8
doorsnede
draagstruktuur
routing
vorm doorbrekend



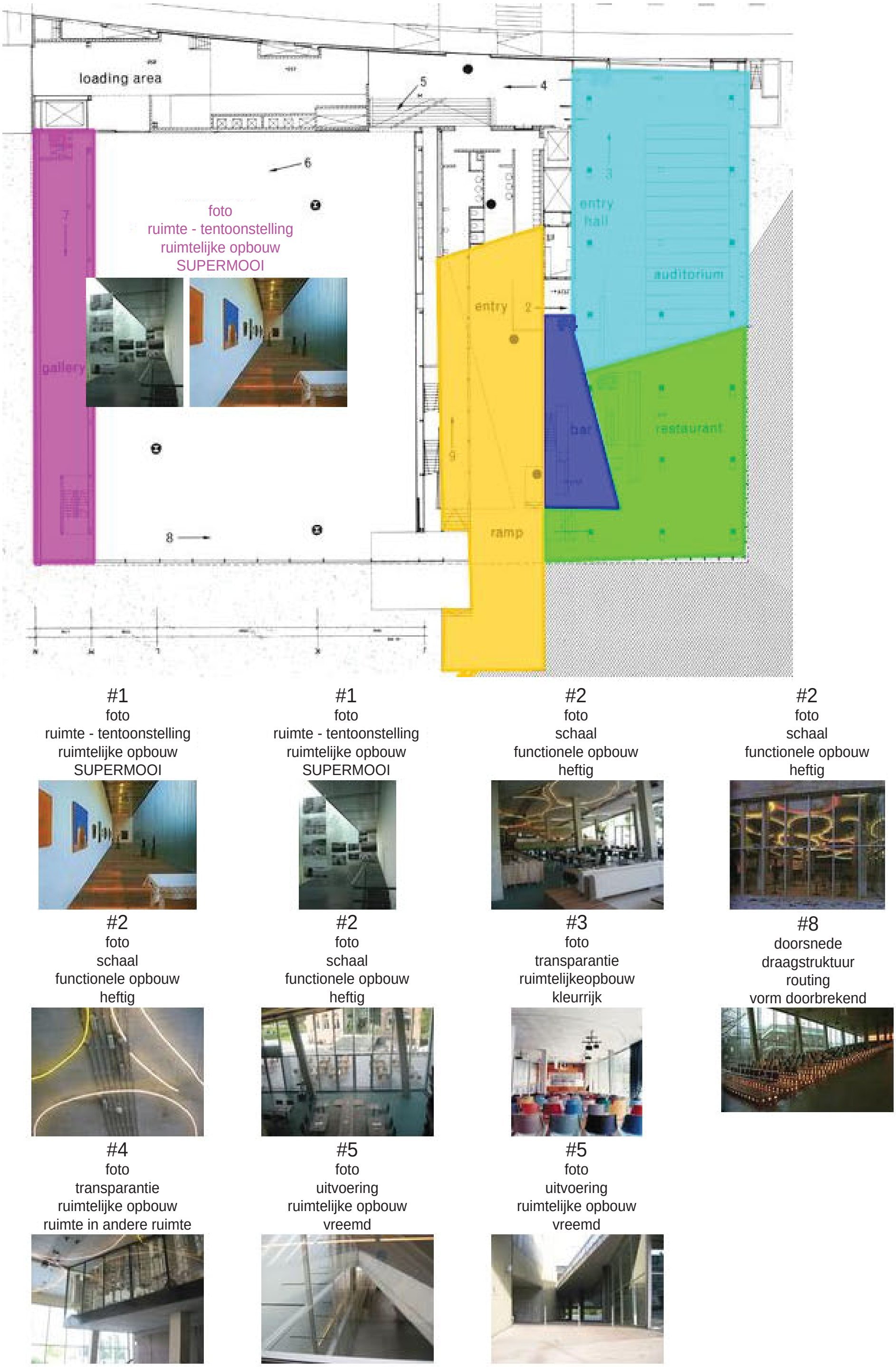


Figure 5.18. The result of the use of the tool is a hyper-image: when the mouse pointer is moved over a marker, the related claims and thumbnails of related documents pop up. If a marker is clicked on, the related document comes up.

main analysis and to the markers, the associated documents and the annotations for each marker.

The result of the various analyses is a common library of analyses that the students can consult for support of their later design activities. A search and browse mechanism is available on the basis of the hierarchy of claims (Figure 5.19). This mechanism is a part of KeySet: KeySet mediates both technically and strategically to ensure communicability of keys. Using the search tool, a student can select any number of concepts/claims, including the key assigned to one's own submission, and retrieve all student works with keys that include all selected keywords. Users can select one or more concepts/claims from each dimension, this selection operates as an AND search, meaning all the terms must be contained in the result. The result set of analyses contains two parts. The first part (top section of the display) contains the analyses where the terms appear in the main analysis. The second part (lower section of the display) contains the analyses where the search terms appear in one of the markers in the analysis. Since the markers and their associated entities can also be retrieved, this ensures that also the contents of the analyses are searchable. The analysis library is a source of information and knowledge for students, instructors, and other interested parties.

The final step in the students' assignment was to search the collection of analyses for examples that support their own design concept for a museum, and write 250 words about how these examples support their design concept. This ensures that they actually explore the collection of analyses, completing the cycle of learning from the analyses.

5.3.4. Reflection on ArcIMap

DAN received highly positive reactions from the design studio coordinator and the design studio instructors. Students were excited about the fact that they could now see the work of all other students. This is usually not the case because of the large number of student groups each semester. In spite of some very promising results and the high enthusiasm of the design instructors, there were a number of problems with the use of DAN in the 4th semester design studio. A detailed and professional evaluation was set up[55] with the goal to find out how an implementation of ArcIMap can seamlessly fit into the educational process, and what the related aspects are. From the use of DAN, it became apparent that students had difficulty to appreciate the added value of using such a tool in their analysis process. It was, however, difficult to determine what precisely the source of these problems was: the shortcomings in the presentation of information and the interface; the functionality of the tool; or the embedding in the education. The evaluation was therefore aimed at investigating the problems with each of these fields (interface design, functionality, embedding in education). The detailed document describing the DAN evaluation and the findings can be found in Appendix C.

[55] In the scope of the 4th semester analysis exercise, the DAN environment was evaluated in a laboratory environment called Laboratory for Work and Interaction Technology (WIT-lab), at the Faculty of Technology, Policy and Management, Delft University of Technology. This evaluation was done in collaboration with Evren Akar and Jelle Attema. The instructors who took part in the evaluation are Steven Steenbruggen and Ernst Janssen Groesbeek. The students who took part in the evaluation are Jilles Berendsen, Carmen Buitenhuis, Mikki Herman, Rosie van der Schans and Sigrun Sumarlidadottir.

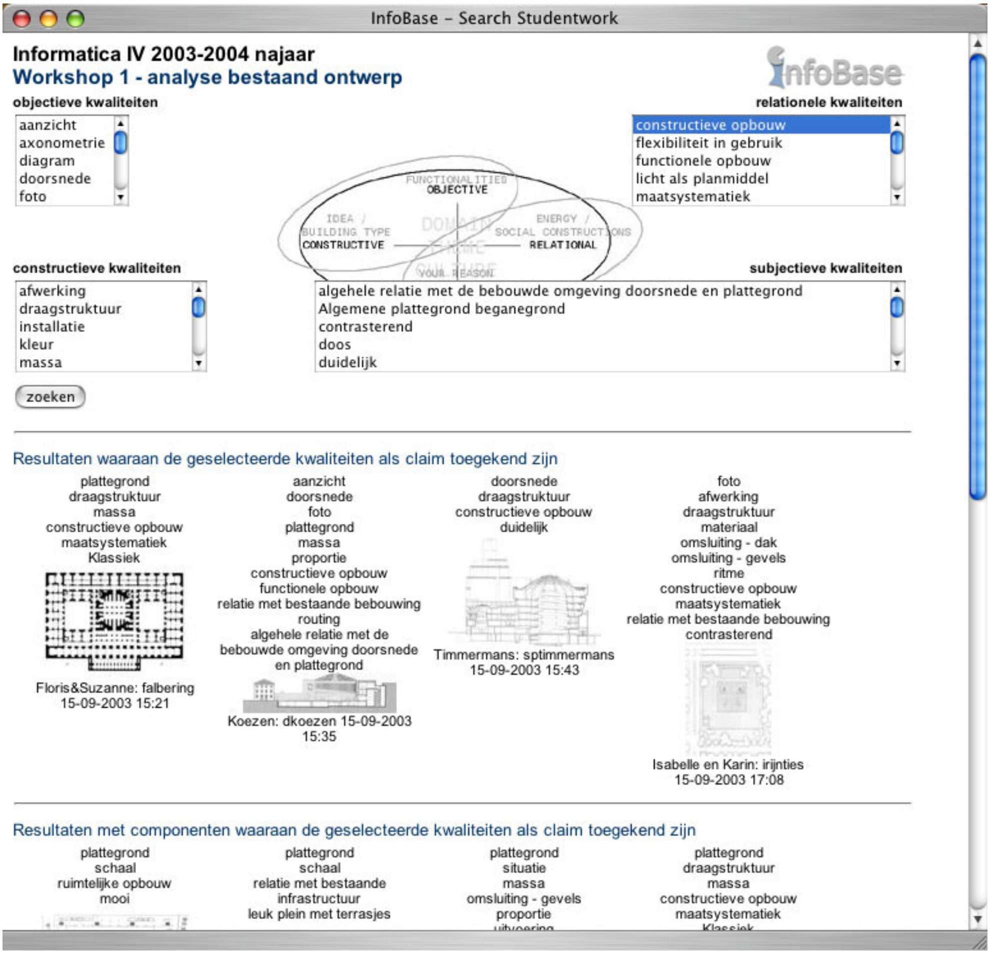


Figure 5.19. The search interface of DAN. Students search the collection of analyses by selecting one or more claims from one or more of the four quality dimensions. The top section of the display displays the analyses where the search term(s) appear in the main analysis. The lower section of the display contains the analyses where the search term(s) appear in one of the markers in the analysis.

Setup of the evaluation: The evaluation consisted of 5 sessions of 2.5 hours each where the author (as course instructor and developer of DAN) sat with an individual student who had already completed the DAN exercise in the design studio in a room with a computer running the software (Figure 5.20). In the control room, the test leader, another project member, and one or two of the other course instructors followed and commented on the activities that went on in the test room. The events in the test room, as well as the comments in the control room were logged in a logbook, and coupled with the video recording of the events in the test room. Each session advanced as follows: the student showed the instructor how she had performed the analysis assignment. The instructor

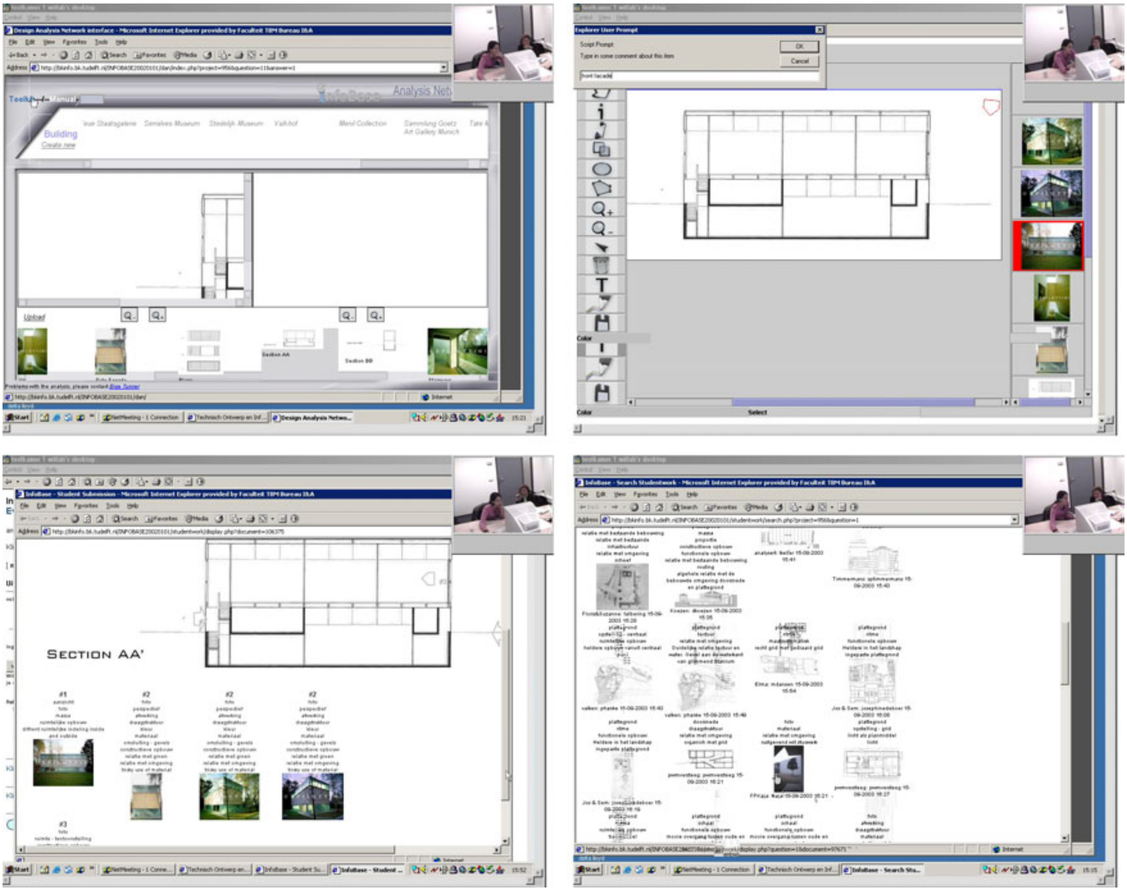


Figure 5.20. Still shots from an evaluation session[56]. The student and the instructor are visible in the top left corner of each image. a) Above, left: The students selects a building and a section from the repository, b) Above, right: The student analyzes the structural setup, which is one of the most characteristic qualities of the building, c) Below, left: The student views the result and comments on it, d) Below, right: The student searches in the database for an interesting example. It is interesting to note that she found the icons representing the analyses dull, and she directly chose to look at the analysis whose icon was a beautiful photograph.

asked for clarification from the student where necessary, or helped with the solution of problems related to the use of the interface. Afterwards the instructor explained to the student what the envisioned role of the DAN tool was during the performance of the assignment. The student then searched the analysis results for a meaningful example in order to illustrate the explanation of the instructor. This example was formulated by the student herself; it was based on her own design ideas. Next, the student performed the assignment one more time using the tool, this time according to the instructor's explanation. At the end of each individual session the activities and comments were

[56] This session was with Mikki Herman.

discussed with the observers and the instructor from the test room, on the basis of the logbook and supported by the video recordings. In these discussions, the goal was to define problems based on the logbook or hinted by the logbook in the areas of ease of use and learning curve of the tool, the added value of using the tool, and the added value and the use of the embedding of DAN in education. The comments that came up in the discussions were categorized and summarized.

Embedding of the application in the analysis process: An important observation was that the precedent library and the tool give too few examples to students (and instructors) that illustrate the added value, and does not provide enough explanation about how students can work with the tool. Furthermore, students have no clear idea how to analyze a building, neither how such a process can run. Moreover not all the instructors have the same opinion about how to do an analysis. When students see how the tool can be used (during the evaluation session) their enthusiasm appeared to be great. In this context, the most important conclusion is that the analysis process must be defined explicitly and clearly and this process must be integrated in the tool. Thus, the tool must assist the users in acquiring a method to do the analysis. Another necessary adjustment is that a number of analysis examples must be provided to the users that illustrate the usefulness and use of concepts. The use of DAN must be truly integrated into the design process, and not be seen by students as "doing double work". Finally, a number of small but disturbing usability problems were detected that hindered the students doing their assignment.

An improved DAN environment can be used for the performance as well the viewing of the analyses. In the development, an intuitive and user friendly interface is a top priority.

5.3.5. KeySet and ArcIMap

As discussed in Section 5.3.2, students learn to lay claims on information collections in order to add value to the information. In DAN, students use metadata along four quality dimensions (Figure 5.15) in order to lay these claims. Students select concepts from all four dimensions in order to form a unique key. KeySet is an instrument and a tool that provides each work with a key consisting of a combination of four or more concepts. KeySet is an instrument that mediates both technically and strategically to ensure communicability of keys (Kooistra et al., 2005). The technical component concerns the search tool of the database where students retrieve work by selecting concepts and keys. The strategic component concerns the implementation of KeySet in education in order to help students learn to deal with the relativity of the information collection, and to use the system to its full extent without a feeling of being left to one's own devices when handling this collection.

KetSet, as it has been developed and used in DAN[57] is beneficial for ArcIMap. KeySet has also been implemented in the InfoBase environment (see footnote 3). InfoBase contains a *StudentWork* interface for students to submit the products and results of their digital exercises and KeySet is integrated in StudentWork. StudentWork and KeySet are being successfully used in the entire B Sc. and M.Sc. education of the Chair of Design Informatics at The Faculty of Architecture, Delft University of Technology by thousands of students

[57] KeySet has been developed together with Jan Kooistra and Rudi Stouffs.

since 2003. StudentWork and KeySet are also successfully being used in the education of the Faculty of Social Sciences at the University of Utrecht.

In the first year of the architectural B.Sc. program, KeySet is used as a closed or almost closed system, i.e., students are limited in their ability to create their own keywords (Figure 5.21). Each dimension can either be completely closed (i.e., a single keyword is provided as a fixed choice), coupled to a fixed set of keywords one can choose from, or linked to an online thesaurus with fixed architectural terms. This stimulates the relatedness of the different keys students use to encode their work while it remains clear that each design is unique and this uniqueness deserves to be honored. Later on in the B.Sc. program the correspondence on claims between designs is primarily dealt with by the students.

The use of KeySet in the B.Sc. education has been extensively evaluated by means of two scales: the Subjective Computer Experience Scale (SCES) and the Subjective E-platform Experience Scale (SEES). These scales are designed to measure the attitude and experience with respect to computer use (SCES) and the use of ICT as work and learning environment (SEES) (Kooistra et al., 2004). Please see Stouffs et al. (2005) for a detailed account on this evaluation.

A questionnaire accompanied the scales evaluation. The results of the questionnaire are demonstrated in Table 5.3. The questionnaire also included an open question: “What did you get from looking at the work of others?” 100 students answered this question. These answers can be divided into five categories: gaining inspiration or ideas, comparing results, (precedent) learning, nothing or not much, and others. 14 students said to be inspired by others or to have gained ideas. More than double indicated to compare oneself with others in terms of pace or level, e.g., “I’m noticeably lagging behind,” “I must work more precisely” or “I saw that others may have modeled more beautifully, but my work was OK.” More than a quarter of the students indicated what they had learned from it, e.g., “I have learned that gothic details exist and that these can form a ‘quadruple joint’,” “I found out that few selected window frames” and “looking how they constructed the different encounters.”

Finally, there were a number of answers that directly substantiated our didactic objectives, e.g., “More insight, because you want to know why exactly they assign specific claims to their detail. You start reading these details better”, “You see how many different types of details can be found with the same keywords”, also “It is difficult to find a reference through the claims because everybody describes their detail very differently” and finally “A feeling of solidarity with my fellow students and curiosity as to what they are busy with.” Even the fact that students had difficulty finding anything still offers clues, e.g., “my detail had a different context, there you go with your keywords” and “not so much, with the keywords I tried to find a detail that was similar to mine and I couldn’t find it.”

The correlations found between the SEES ICT and KeySet factors and the variance analysis conducted has clarified the strategy that needs to be followed. The more students are familiar with dealing with metadata (KeySet), the more they will find it worthwhile, and also rather fun. The latter not only depends on whether the instrument is profiled appropriately but also on the courses or workshops in which it is included. As such, it also depends on a stimulating policy of the organization it is embedded in.

Figure 5.21. StudentWork application of InfoBase where all Design Informatics work in the Faculty of Archecture, TU Delft gets collected in an interactive and publicly accessible database. StudentWork utilizes the KeySet tool.

Table 5.3. Results from the short questionnaire supplementing the scales evaluation. 117 students completed this questionnaire.

Question	Percentage of answers			
	Yes	**Somewhat**	**No**	**-**
Was the instructor's explanation clear?	39%	38%	17%	6%
Do you understand what is meant with each of the four dimensions?	55%	33%	7%	5%
Do you understand why these metadata must be assigned to the design?	54%	25%	12%	9%
Were the metadata useful when searching for work of others?	33%	33%	22%	12%

KeySet is beneficial for ArcIMap in its use for novice architecture students because it provides a philosophical organization of four dimensions to the semantic structure of ArcIMap. Hence KeySet can be considered as an instance of ArcIMap. StudentWork and KeySet are also open systems that go on living through their use by large numbers of students as similar for ArcIMap.

5.4. DESIGNMAP: AN EXPERIMENT IN ARCHITECTURAL PRACTICE OF ARCIMAP

DesignMap was developed in order to test the validity and applicability of ArcIMap in a practical context. DesignMap was applied in an architectural office, Mecanoo[58], located in Delft, The Netherlands. The office has about 65 employees, consisting mostly of architects and engineers.

Because of recent advances in Information, Communication and Knowledge Technologies (ICKT), architectural offices are going through a process of digitalization. It is becoming increasingly common practice for architectural offices to digitally archive their project documents and to set up intranet sites for their employees. In this respect, offices are looking for organizational structures for their precedent documents that are suitable in order to support an easy and fast but, also, effective retrieval of their documents. This was also the case at Mecanoo.

The design flow at Mecanoo has three main stages.

1. Idea development stage
2. Design stage
3. Execution stage

The idea development stage starts with meeting up with the client and identifying the project requirements. This is followed by visiting the site of the project and getting a feel to the size and environment of the project. Then, brainstorming sessions are held with teams from different projects to set up a number of plans for the project. One of these is chosen after a discussion with the client. The design stage starts by setting up a team of architects to perform the design process. The chief architects in Mecanoo have input on all projects and give the team freedom to have different ideas. The design stage progresses like a wave: it begins with a lot of input information (viewing lots of pictures, for example), then gradually the idea gets fixed and the design is completed. The execution stage is concerned with the actual construction of the design. Mecanoo distinguishes itself as an architecture office by getting involved in both the design as well as the execution of a project. During the execution of a project, more technical and less architectural expertise is needed, as opposed to the design stage of a project, where more architectural and less technical expertise is needed.

Designers at Mecanoo look at magazines, and use search engines such as Google to find relevant information for the design project at hand. They organize documents on a server according to a naming system used throughout the entire office. However, when people use a document or image, they tend to copy it to the hard disk of the computer they are using. This results in many copies of images on different computers and this causes problems of too little hard disk space. It was told that before the office switched to using computers in design, there were drawers with stickers of drawings, such as trees. Designers would take these stickers and apply them on paper. After switching to computers, employees started making copies of images. This imitates their previous behavior into the

[58] http://www.mecanoo.com/. Our contact people there for this project were Nick Marks (project manager, engineer) and Iemke Bakker (senior architect).

electronic environment. Furthermore, since this classification system is very rigid, they have great difficulties reaching documentation of their earlier projects.

In this particular office, a lot of aspects from previous designs are reused; this is an office policy. Images and documents are stored on servers, but there is very little recorded information available about the design rationale and crucial concepts. Furthermore, there is a single rigid classification system that does not allow designers to use subjective terms.

DesignMap[59] is a web-based collaborative environment that implements ArcIMap and is a flexible and extensible system intended to be used at the early stages of design. It targets middle and small-size architectural offices. The DesignMap application has three main goals:

- to enable a design team to organize the information that they gather during the conceptual design phase in a personal, flexible and extensible manner,
- to enable a design team to build up a common language of design concepts and relationships in order to improve their communication,
- to enable the recording and reuse of design knowledge generated by a design team.

Keeping these goals in mind, the main motivation behind the development of DesignMap is that such a record of the creative thought process makes it possible to accelerate and improve the quality of the design process of a given design team. It also enables the comparison between the methods different design teams use to design a given product. This, in turn, opens the door to learn from the methods different teams employ to design their products. This also makes it possible to evaluate the strengths and weaknesses in the design process of a design team, and subsequently rectify any observed weakness. Finally, it enables identifying the relationship between two different aspects of the product, by tracing the way these aspects have been developed in the design process. This, for example, makes it possible to evaluate the impact of modifying a specific design aspect on the rest of the product.

Accordingly, some of the requirements such an application needs to fulfill are:

- The system must incorporate a way to visualize ideas and thought processes from different users into a common framework.
- It has to enable multiple users to access a common database, where they can retrieve and contribute to the current framework of the design process.
- Multiple users must be able to access and modify the database concurrently, without introducing errors or inconsistencies into the system.
- The system should be easy to use, and needs to have an intuitive interface so that users can focus on generating ideas rather than controlling the system itself.
- The interface has to be easily extensible and flexible, to suit the different design processes adopted in different disciplines.

[59] DesignMap shares some development (database, scripts) with the InfoBase project.

5.4.1. The DesignMap application

The first prototype implementation of Design Map offers a mechanism for a group of users to build up a network of concepts and conceptual relationships that reflect the common working processes of an architectural firm and its own design processes. Furthermore, it enables the classification, archiving and retrieval of multi-media design documents, using the semantic structure as the organizational backbone. Being a web-based environment, DesignMap enables users to record and exchange their ideas about a design project independently of any space and time constraints. During the design process, there is a need for an intuitive and flexible method to register the important concepts at different junctures and inform the team members of these concepts and how they relate to each other, and how they are represented in the design project, and to transform this knowledge further to new members. Even after the design process ends, it is sometimes beneficial to evaluate the impact of various concepts used by the designers in earlier stages of the project on the overall progress of the design process.

The input to DesignMap is a semantic structure and a number of design documents. The semantic structure is defined by a group of users. Users have equal access rights: this makes it a completely democratic system. Users can add to or change the input of other users. An initial semantic structure was created by the project manager[60] from Mecanoo Architects (Figure 5 22). This list contains physical concepts as well as abstracts concepts such as 'inspirations'. Still, one can tell that the main line of thought behind this classification is the archiving of documents rather than a dynamic communication and organizational structure made up of concepts and relationships.

The main interface of DesignMap (Figure 5.23) contains functionality to upload documents and search and browse documents by concept. Additionally users can modify and delete documents and their properties. The bottom part of the interface is similar to the bottom part of the Analysis Presentation tool described in Section 5.1.3 and initially shows all documents in the system by displaying their thumbnails. Above each thumbnail is its title. Below each thumbnail is a list of the associated concepts as hyperlinks. Clicking on a concept displays the documents associated with that concept. This allows for associative browsing of concepts and documents. This bottom part of the interface also displays the currently selected concept(s). One can drag and drop thumbnails into the two areas above to view them in a bigger size. There is also an option under each of these compartments to view the document in a separate window in full size. One can interactively zoom in and out the image documents. The 'search by category' option at the top of the interface brings up the category view window, allowing the user to select one or multiple concepts for searching documents[61]. The associated documents are presented in the main interface. Documents of all formats can be uploaded to DesignMap. Thumbnail images can be provided for documents that are not images. When a document is uploaded, the system requires the user to select at least one concept from the concepts hierarchy. Users can create new concepts in relation to the document that is to be uploaded. Once a document is uploaded, it is visible to everyone using the system. One can modify or delete a document and its properties by double clicking on its thumbnail.

[60] The project manager is Nick Marks.
[61] This is an AND selection.

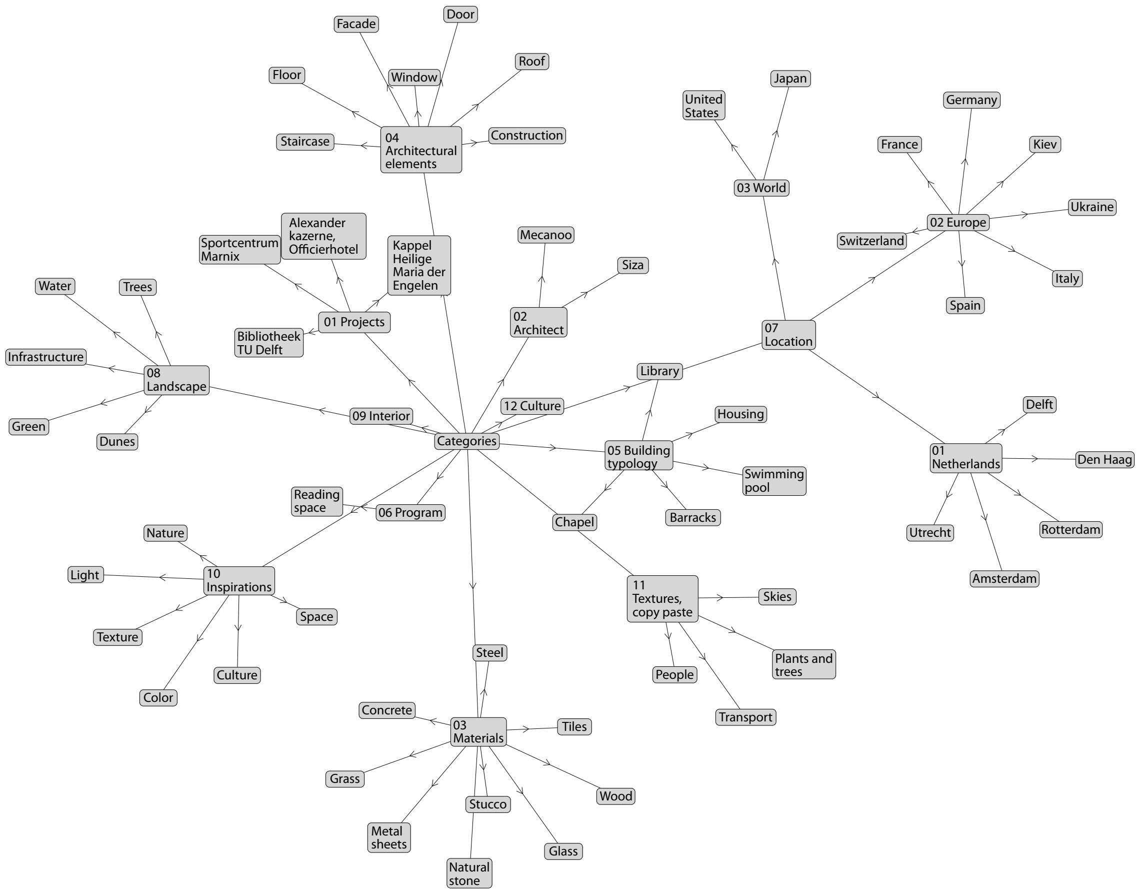


Figure 5.22. The semantic structure defined by the project manager at Mecanoo.

After an evaluation session with architects at Mecanoo a second prototype implementation of DesignMap was implemented which improved on the first implementation in order to foster the usability of the application. The new DesignMap has been implemented as a Java application to ensure usability across different platforms, as well as flexibility in terms of tool extension and adaptation. On the backend, DesignMap stores data in a MySQL database through a PHP interface[62]. Communication with the database and between different DesignMap clients is achieved using a set of XML messages sent within a local area network and across the Internet. In addition to the requirements stemming from the targeted usage of the tool, there are other requirements related to the software environment the tool is supposed to connect to. Figure 5.24 shows an illustration of this environment, where DesignMap is shown to connect across the Internet to a shared database located behind a firewall. The three main interacting parties shown in the figure are the 'user side', the 'internet' and the 'database side'. The user side

[62] This is the InfoBase structure.

Figure 5.23. The main interface of the first DesignMap prototype.

connects a number of DesignMap clients with each other using a local area network (LAN), which allows these clients to communicate and share a common design environment. The data generated and stored by these clients are located in a database on the database side of the figure. The communication between the clients and the database takes place over the Internet, and has to be filtered thorough a firewall. This requires the communication with the database to be carried out through an HTTP interface, which is achieved by the application clients in XML.

A web server (or HTML server) on the database side processes the requests issued by the DesignMap clients and forwards these to a MySQL server which, in turn, responds to these as appropriate based on the information in the database. The user side is represented by the members of a design team in a company or in a university, all of whom are using the DesignMap tool and are connected together with a LAN. The view of the information structure these clients display is synchronized at all times. The database side, on the other hand, is represented by the InfoBase database infrastructure.

A screenshot of the graphical user interface of the application is shown in Figure 5 25. Besides the menu bar and the status bar, the tool has three main panels: the tree browser, the dynamic network browser and the thumbnail browser. The tree browser and the

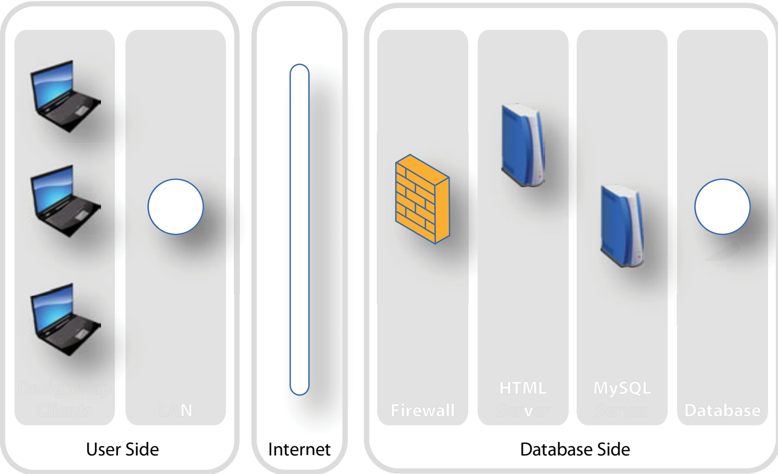


Figure 5.24. Illustration of the software environment of DesignMap.

dynamic network browser in the figure offer two different ways to represent the semantic structure. The tree browser presents the semantic structure in the form of a clickable tree. The dynamic network browser allows the user to navigate around semantic structure in the form of a traceable network[63]. These two browsers are synchronized. Each time a concept is highlighted in the tree browser or the dynamic network browser, the related documents are shown in the thumbnail panel. Clicking on the thumbnail of a document opens up a popup window that shows that document.

A description of the system architecture of DesignMap is included in Appendix D.

5.4.2. Reflection on ArcIMap

In general, highly positive reactions from the participants at Mecanoo were received about the use of this environment in their office. The users especially appreciated the fact that the semantic structure can be modified and extended without affecting the already stored documents in the environment. This experiment ran for approximately 3 months. Because of the heavy work load of the designers and project deadline constraints, and because of financial considerations, it was treated as a pilot project and was intensively used by only two individuals in the office: the general project manager who tested the system and prepared the semantic structure, and a designer who input documents. After an initial period of use and coaching, in order to evaluate the prototype and its use, a 4.5 hour workshop at Mecanoo was conducted. The participants in this workshop were the design and development team of DesignMap[64], the project manager from Mecanoo and five architects from Mecanoo. The participants from Mecanoo had received the agenda of the workshop beforehand and had prepared for it by using the application.

[63] The dynamic network browser has been implemented using TouchGraph, http://www.touchgraph.com/
[64] Bige Tunçer, Evren Akar, Zaid Al Ars, and Joost Beintema

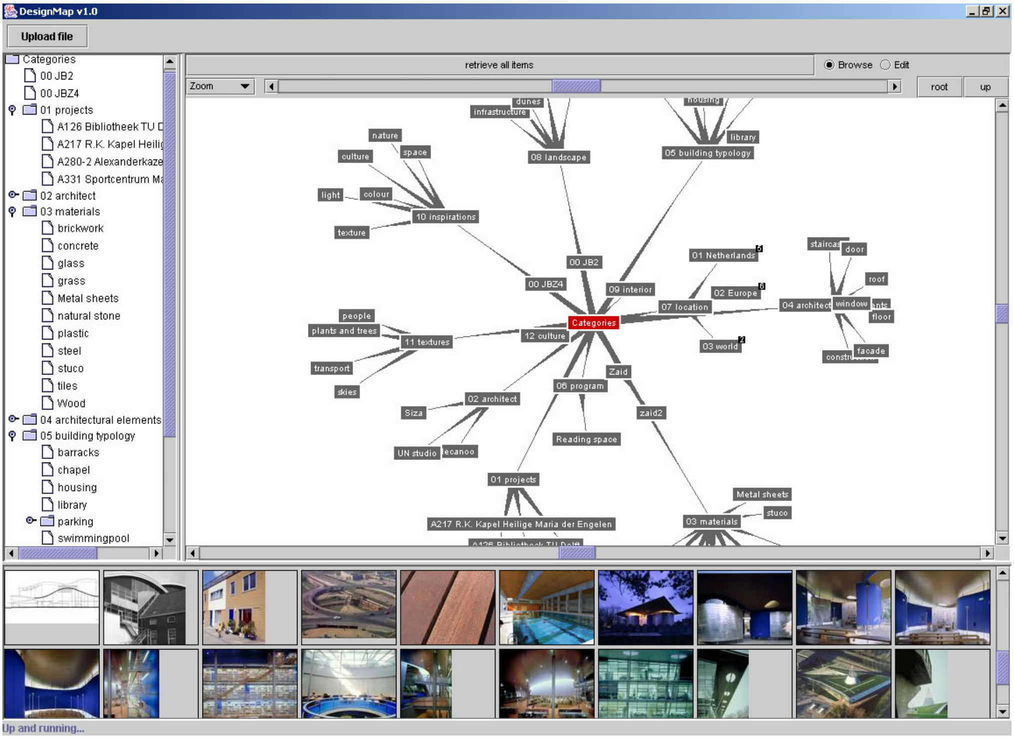


Figure 5.25. Main window of the DesignMap, where the different components are shown.

The workshop consisted of three main parts. It started with a presentation by the author, the main designer and developer of DesignMap, describing the program, its functionality, its intended use, and its goals. There were prints with snapshots of the interface hanging on the wall (Figure 5.26). Next, there was a presentation of the general design process flow at Mecanoo.

The second part of the workshop consisted of a brainstorming session on how the DesignMap application can be embedded in an active project at Mecanoo. Participants used notes to write down ideas and put them on the wall near another idea that is related another idea (Figure 5.27). Later, these were further grouped together, themes were derived and discussed.

The main conclusions of this brainstorming session were that DesignMap can be applied in two different ways:

As a structured information repository: Currently, Mecanoo designers use a special directory tree structure to organize files on a server, and a web search tool such as Google as a way to retrieve additional information and store these on their own computers. This approach is familiar and easy to use for them. However, it has disadvantages. There is no cognitive support for searching or browsing of the stored information. Additionally, there

Figure 5.26. The presentation of the first DesignMap prototype during the evaluation and brainstorming workshop at Mecanoo architects.

is only one way of looking for information: traversing the directory structure. Additional relationships are not supported which would assist designers in their design process. The most important requirements that came up for a system that supports designers by providing a structured repository were ease of use and flexibility.

As a means for correspondence: Currently, there is no such system being used at Mecanoo, or any other office they are aware of. The architects seem to be very much interested in such an application.

The workshop ended with a discussion and conclusions. One of the conclusions was that the system should be used both as a structured information repository and as a correspondence system.

Designers did not initially show much interest in DesignMap, because they did not understand its potential as a correspondence tool. Designers are not interested in a predefined keyword structure that they need to abide by. Designers were initially introduced to the concept hierarchy that their manager had constructed, and they thought that this was a fixed structure that they needed to abide by in their design process. This was not very interesting for them and they did not see the added value of this in the design process. On the contrary, they very much liked the idea of a concept network that they design collectively that reflects their thought processes in the conceptual phase of design. Once they understood the added value of the approach, they became highly enthusiastic. The project manager, on the other hand, was not as excited about the use of the tool as a correspondence tool. What he needed was a ‘rigid’ classification system for use in the whole office. The gap between designers and managers in design offices about expectations of design aid tools seems to be a widespread issue.

The results of this evaluation workshop included some points categorized below:

Figure 5.27. Some examples of the outcomes of the brainstorming session during the evaluation and brainstorming workshop at Mecanoo architects.

Semantic structure: In order to decrease the cognitive load of designers while building up a semantic structure, concepts in the structure should semi-automatically match each other. Additionally, in order to fulfill the use of DesignMap both as an archive environment and as a communication tool, two types of semantic structures are needed: one specific to each project and/or design team, and a filtering mechanism that filters all the structures and fits them into a mediated archival classification system.

Uploading of documents: Ease of use is very important in terms of uploading documents into the system. One should be able to upload a number of documents together. For example, a folder with pictures having the name 'inspirations' can be automatically uploaded and all files in the folder assigned the concept 'inspirations'. Another example is to define an area (such as a button) with the label 'inspirations' and be able to drag and drop documents into such an area in order to upload them. This will also assist with initially filling the system in the least cumbersome way with precedent information allowing for immediate browsing and searching at the beginning of a project. For the purpose of initially filling the system, a web-crawler-like program can run through the computer and generate a categorization from the existing directory structure and upload the documents.

User interface and interaction: The interface must be very visually oriented and intuitive to use in order to be easily adopted by designers. A drag-drop interface to relate a document with concepts would be easier. Two areas for viewing full size documents are not sufficient. A post-it note like interface for viewing documents would be interesting.

Embedding in an organizational structure: An application of ArcIMap for an architectural office must be developed taking the work processes of the office into account. The application developer must base the application design on research into the needs of the designers. Once an application of ArcIMap is adopted within an office, its use must be iteratively evaluated possibly through questionnaires and interviews.

5.5. CONCLUSION

The added value of using ArcIMap in conceptual design processes has been validated through the four applications in different phases of education and in a practical context. The two most important conclusions are that users need to learn how to work with ArcIMap, and that an application of ArcIMap must be tailor made for its use context in order to be successfully used.

ArcIMap defines a framework for collectively creating information structures and corresponding over the information. This framework does not force any standardization in any way on its users. However, the users need to learn to express thought processes in semantic structures using the methods of concept mapping. There is a learning curve for users when starting to use applications of ArcIMap. Therefore, users of ArcIMap applications must receive training at the before starting use such an application and during its use.

An application of ArcIMap, in education or practice, must take the existing working processes of its intended users into account already in the design stage of the application. This requires the designers of the application to research the working contexts of intended users, formulate application functionality requirements accordingly, and get the opinions of the intended users possibly in (application) design workshops. A participatory application design process is needed where the designers, developers, and users of the application come together in order to make a successful application of ArcIMap.

6

CONCLUSIONS AND FUTURE WORK

How can communities of architectural practice correspond on design information and knowledge during the conceptual phase of design? This was the main question that was tackled in this research. It has been addressed through a study of relevant literature, theories, methods and techniques.

Members of a community of practice share a domain, interact with each other and learn from each other (Bowker and Star, 1999: 294). They cooperate and generate a common understanding and common knowledge both by recording knowledge into documents and by actively participating in social processes in order to personally contextualize this recorded knowledge (Wenger, 1998). A community of practice thrives only if these two activities coincide. By performing these two activities, members of an architectural community of practice correspond on the common information and knowledge, collectively agreeing on the value of this knowledge and information. These processes are both the means and the result of an architectural community of practice.

The information collection that an architectural community of practice corresponds on in the conceptual design phase is a collection of design documents that designers use to gain knowledge and inspiration. Designers' cognitive and information retrieval needs in this phase vary: a known specific document, or documents pertaining to a certain concepts, or just randomly browsing and following suggested links. In order to support such needs, a complex information structure is necessary that is composed of information entities and their relationships, tagged with certain design concepts in order to be easily retrieved. These design concepts are themselves related through semantic relationships forming a semantic structure. This semantic structure acts as the organizational backbone of a complex information structure. Elements (concepts and relationships) of this semantic structure are associated with information entities (documents) and describe them. This allows and supports *associative browsing*, where users browse using the underlying associative relationships between information entities. This enables cognitive jumps and unexpected creative discoveries.

The final outcome of this research is a computational framework – ArcIMap – that encapsulates the above mentioned phenomena, and enables the definition and implementation of specific applications in various contexts. Complex information structures form the basis of ArcIMap, which defines a structure for the design and creation of digital applications that support designers in the conceptual phase of design. The goal is to define the representational framework for achieving an integrated information structure of components, relationships and metadata from a collection of design documents and the knowledge that resides in these documents. The framework can be implemented for different purposes, domains, and contexts. Four prototype applications of this framework were developed and tested in different architectural education and practice contexts.

This final chapter discusses the results of the research presented in this dissertation, identifies the main contributions of this research, and establishes an agenda for possible future research directions and steps in the field of information organization and knowledge representation in architectural design.

6.1. THE ARCIMAP FRAMEWORK

Within the conceptual phase of design, architects (and designers in general) collect and look at design documents. These design documents are sources of knowledge and inspiration for the designers. In architecture, these documents often represent precedent designs. A precedent is a specific building that entails a specific solution to a problem. An electronic information organization environment for the collection and organization of these documents is useful both in the educational and the practical contexts. Furthermore, in order to develop design expertise and skills, the knowledge that resides in these documents provides an invaluable source. A system that can be used to organize, structure, and reuse this knowledge is a necessary tool for any architect. In this context, the main result of this research is a computational framework, named Architectural Information Map (ArcIMap), to be used by a community of designers during the conceptual phase of design for information organization and knowledge representation.

ArcIMap can be considered as a framework for the design and development of specific computer applications for information organization and knowledge representation in design. This is demonstrated by the breadth of the working prototype applications of ArcIMap described in Chapter 5: an educational architectural analysis environment, an educational cooperative knowledge base that supports the design of double curved surface buildings, and a design information organization system for small and middle sized architectural offices have been built using this model. ArcIMap can be implemented for any design context where there is a need for information organization and knowledge representation. ArcIMap does not make claims about the content of the information and knowledge that it models. This content is decided upon and created by the users of an application of ArcIMap. However, ArcIMap defines two main structures: the semantic structure, which is also the information organization structure; and the document structure, which is the collection of documents and their decompositions. The situating of an application in its context is also an indispensable component of ArcIMap applications, which is the work context and organizational processes of the intended users. The applications that implement ArcIMap are bound to contain these structures.

The semantic structure is defined as a semantic network made up of concepts and relationships. Methodologically, it is based on concept mapping that has its roots in learning theories. Formally, it is based on graphical knowledge representation formalisms, specifically semantic networks, conceptual graphs and topic maps. In its definition, it is a semantic network. The concepts and relationships are used for representing and structuring knowledge. The relationships between concepts are semantic relationships, which can represent associations between concepts. These associations can be highly personal, therefore allowing the subjective definition of knowledge. This knowledge structure can represent declarative knowledge, and through the use of associations and the subjective definition of concepts, it can represent procedural knowledge. This is a "subject-based" use of the knowledge structure, meaning this knowledge can exist independently of the design documents stored in the system.

The document structure concerns the collection of the design documents contained in the system. These documents are indexed using the knowledge structure; each document is related to a concept or a relationship in this structure. Concepts that do not yet exist in the knowledge structure may emerge by looking at the documents; then these can be added to the knowledge structure. This is an "object-based" use of the knowledge structure; content related properties of documents (or metadata) can be incorporated into the organizational structure.

Syntactically, documents can be broken up into components. For example, if there is a collection of documents for the building type theaters, one may wish to specifically look at the foyers in these theaters. Through a specific marking of the foyers, where this marking is also related to a concept or a relationship in the knowledge structure, one can retrieve specific components and compare the issues of interest. This feature supports the acquisition and explicit definition of knowledge that resides in design documents. In this way, documents are indexed by their content, rather than just as whole documents. Additionally, documents, and document components, can also be referenced by each other, and these references related to an item in the knowledge structure. This indexing of document components enables an outsider to access this information more effectively, independent of the viewpoint of the person who conceived it, i.e., the document's content and composition. First, it allows one to access specific information directly instead of requiring a traversal of the document hierarchy. Individual components can be reached and retrieved more quickly when indexed directly. Second, components can be considered from a different point of view. The location of a component in the structure is no longer only defined by its place in the document hierarchy. Instead, components provide direct access to other related components, forming a part of the first component's view. Third, one can access the information structure from alternative views to those that are expressed by the individual documents. One can define design stories by relating documents and indexing specific components.

The situating in a specific environment concerns the context, users, and information processes. The design of an application of ArcIMap must consider the context in which this application will be used. The potential users, depending on their experience levels as designers, may have different needs. The work processes of the organization and users must be integrated in the design and interaction of the application. Therefore a participatory process is needed in the software and interaction design of the intended application. This requires a study of the users, their professional context, and a study of work and interaction processes in the form of interviews, observations and discussions within the use context of the application, and a translation of these into software requirements. Before embedding the application in its use context the users of the application must also be enlightened about the application, its purposes, and the goals and premises of the underlying model.

The creators of the information and knowledge are also its users. The ArcIMap model doesn't prescribe any specification of access rights for its users. Depending on the purpose of the application, it may serve a completely democratic system where all the users have equal access rights, or a system where some users have more rights than others. For example, in the case of an application used in the educational domain, an instructor that sometimes needs to correct or change input will have more access rights than the students using the application. But in the case of an information organization environment being used in an architectural office, the members of a small design team may have equal rights for democratically building up a knowledge structure and inputting and relating documents within the system.

ArcIMap is a framework from which applications for design aid can be developed. It has been conceived and implemented for architectural design, but it can be usable in more design fields than architecture. It can potentially be applied, among others, in industrial design, interaction design, and software design, although further research should demonstrate the validity of this statement.

Applications of ArcIMap are open systems that house digital information structures created by communities of practice. The prototype applications of ArcIMap that have been developed and included in this dissertation demonstrate this quality. Specifically, the StudentWork interface of the InfoBase environment that integrates the KeySet tool, which is an instance of ArcIMap, is effectively being used in education by thousands of students each year in two different universities since 2003. This use demonstrates the success of ArcIMap as a method and as a model, especially in education.

6.2. ARCIMAP IN DESIGN EDUCATION

In many design studios, students are asked to perform an analysis of one or more existing buildings related to the building type of the design project of the studio. They usually do this in the form of collages on paper, or as collages using an image processing application. It is a common concern of design instructors that students sometimes approach the analysis as an action of highlighting some parts of a building, without really thinking about the underlying design concepts. Furthermore, when the students do not provide an explanation for these highlighted areas, these analyses have very little meaning. As the Design Analysis Network (DAN) application demonstrated, when students are faced to take the time to explicitly state what is the purpose of an action, by providing a claim for that action, their analysis results gain quality. More importantly, they learn more from the design precedent that they analyze. The fact that students must also provide their personal criticism for the analyzed precedent also adds to their design thinking development. Thus, design analysis in education stands to benefit majorly from a prominent application of ArcIMap.

A second application area in education is knowledge modeling and exchange. In the architectural education in Delft, it is commonplace for students to do role-playing in a team: for example, one student takes on the role of the architect, another the structural engineer, another the project manager, etc. This is usually done as part of a multidisciplinary design project. Each of the students needs to acquire knowledge about her specific domain. Furthermore, they need to share this with the other group members, because the final aim of this education is for all students to have knowledge and skills in all the involved disciplines. When students can record the knowledge involved in a domain and they can collectively define associations between entities in domains, this assists in their learning process. This has been demonstrated by the Blob Inventory Project (BLIP) application.

The evaluation of these two working prototype applications clearly shows that the use of the application must be embedded in the context, and that the process of design in this context must be taken into account in the intended use process of the tool. Also, students need to be provided with many examples before they start using the system; instructors must actively be involved in the use of the system. Additionally, students should not feel they do double work, the submission and evaluation of assignments must be achieved through the use of the system.

Another important point is that the creation of the knowledge structure is based on the method of concept mapping, which is built upon the assimilation theory of learning. This cognitive psychology theory states that "learning takes place by the assimilation of new concepts and propositions into existing concept propositional frameworks held by the learner" (Novak, 1998). Concept maps are built up by creating a concept and its relationships one at a time, slowly adding to the concept propositional framework of the user. This supports deep learning, where the content gets stored in the long term memory of the student. This in turn contributes to the building up of design knowledge.

6.3. ARCIMAP IN DESIGN PRACTICE

It is a challenge to apply the results of this research into practice, where the 'time is money' syndrome is widespread, and where some immediate profit is generally expected from the use of such a system. In our conversations with professional designers, the first obstacle in the acceptance of such a system is distinguishing the use of such a system from the use of magazines or web search engines. Also, designers are not interested in a rigid classification structure that they must adhere to in the organization of their documents. However, the enthusiasm level of the designers highly increases once they realize that they can model and enter their own subjective concepts into the semantic structure and that they can freely organize their documents. By defining designers as active organizers of the knowledge and inspirations in a design context, the focus shifts from a pre-structured use of design precedents to an environment that facilitates and encourages the designer in creating her own structured body of design knowledge. The evaluation of the working prototype application DesignMap, which was used at Mecanoo Architects in Delft demonstrates this.

Designers appreciate the fact that ArcIMap allows for a flexible and extensible definition of information and knowledge structures. Within a practical application, when they need to modify or change the knowledge structure, the document structure is not directly affected, and vice versa. This feature saves a lot of time and concern for the designers. Additionally, the fact that ArcIMap allows for the indexing of parts of documents is also appreciated by designers. They can focus on certain knowledge in a document, and index the part of the document that contains this specific knowledge. This is very practical and efficient for them.

Such a system can be used for two purposes in an office: in order to create a common knowledge structure for use in the entire office, and in order to aid the design process of small project groups by having them cooperate through temporary project-based knowledge structures. Project managers are generally highly interested in the first use, and designers in the second. In order to transfer knowledge and information from such temporary structures to the common knowledge structure, ArcIMap foresees in translation and mapping mechanisms. This enables ease of use with respect to the wishes of designers as well as managers.

There is an enormous time and financial pressure on the designers working in architectural offices. The uptake and use of such systems require an investment from the designers, as well as from the managers. Designers and managers need to be convinced of the advantages of the use of such systems in the long term, and not immediately reject their use because they cannot see an immediate short term profit.

However, there are a number of improvements that would increase the chance of success of an application of ArcIMap especially in the context of its use in an architectural office. One important requirement for an information system to be used in practice is that it should have as little impact on daily work as possible. This concerns the user interface; it needs to be as user friendly as possible, and the user interaction needs to be very intuitive for architects. Additionally, the main problem concerning attempts to capture and make available knowledge during design is the additional workload that is generated by the capture process and how one can justify this to designers and managers. In order to decrease this workload, a number of tools must be developed and implemented. These are elaborated on in the next section.

6.4. FUTURE RESEARCH DIRECTIONS

The ultimate assessment of the ArcIMap model requires the evaluation of an application embedded in a context. Since ArcIMap in its entirety has not been implemented within an application, a conclusive evaluation remains open for future research. However, before such an evaluation, a number of improvements can be mapped out that will facilitate the use of an application of ArcIMap, especially in a professional design office:

Automatic generation of a taxonomy from collected documents in order to make the use of the system less cumbersome at the beginning. Once this taxonomy is generated, it can be used as the initial organizational structure for indexing documents. This is a more sophisticated version of the use case described in section 4.4.3, "Automatically filling the system from a file structure". In order to automatically generate a taxonomy, Analytical Hierarchy Process (AHP) can be used to dynamically cluster concepts into a hierarchy (Saaty, 1980). The AHP process is a technique to compute the priority vector, ranking the relative importance of factors being compared. AHP uses qualitative data and deals with uncertainty, imprecision and subjectivity. The only inputs to be supplied are the pair wise comparisons of relative importance of factors, taken two at a time.

Semi-automatic capturing of knowledge from design documents using pattern recognition mechanisms in order to enable easy indexing of documents. The process of document indexing and decomposition may be (semi-) automated using pattern recognition mechanisms and artificial intelligence techniques. Image recognition mechanisms for images (e.g., Barrow and Tenenbaum, 1981; Koutamanis, 1995), shape recognition mechanisms for simple line drawings (e.g., Chase, 1989; Krishnamurti, 1981), and keyword or concept recognition mechanisms for texts (e.g., Greenberg, 1999b) can assist in presenting the user with suggestions about document components corresponding to a given semantic structure. These have been fleshed out in Section 3.3.2. Other formats require similar, though different, recognition techniques. However, more research is needed in order to provide a satisfactory method and developments in order to make a significant contribution in this point.

Connectionist neural networks should be combined with conceptual structures to get the best out of both domains (Way, 1994). Working at the knowledge level of cognition is highly suitable for application in conceptual design, however, ontological engineering by using conceptual knowledge representation has its disadvantages when the amount of knowledge to be represented gets large. Connectionist approaches and conceptual approaches should be used in a complementary manner, using the strengths of both. The main future research area is the development of "hybrid systems that can use the power of each to create more flexible and robust systems" (Way, 1994: 21). In this context, the process of document decomposition can be (semi-)automated using pattern recognition mechanisms of AI techniques.

6.5. REFLECTION ON THE EVALUATION OF ARCIMAP APPLICATIONS

The four prototype applications of ArcIMap have researched, evaluated, verified and validated certain aspects of ArcIMap. The evaluation of the model in its entirety has been achieved by assessing the model's compliance with the requirements. However, the ultimate assessment of the model requires the evaluation of an application embedded in a context. ArcIMap in its entirety has not been implemented within an application. Therefore, a conclusive evaluation remains a topic for future research.

An evaluation of acceptance of use of an application in a context can be done using the Technology Transition Model (TTM) (Briggs et al., 1999; Briggs et al., 2001). TTM has been developed on top of the Technology Acceptance Model (TAM) (Davis, 1989). TAM claims that a new software is used if that software is perceived as useful and easy to use. However, TAM provides little explanation as to why a certain uses (or users) may (and usually do) find an application easy to use and others find it impossibly difficult to use. TAM and TTM both claim that system use (U) is a positive function of behavioral intentions (B). TTM builds up on this by declaring that the 'intention to use' is a multiplicative function of the perceived magnitude (M) of the net value that might be obtained after a switch to the new technology, and the perceived frequency (F) of the obtained net value (Briggs et al., 2001). The causal relationship between M*F and B is moderated by degree of certainty (C) of the usefulness, which is strengthened by exposure, and the perceived net value of transition (T) to the new technology. According to TTM, perceived net value (V) is an overall sense rather than a rational summation of cost and benefit. When new software is being considered, value is perceived along the following dimensions:

- "*Affective:* The extent to which the technology will invoke positive or negative emotional response in the user.

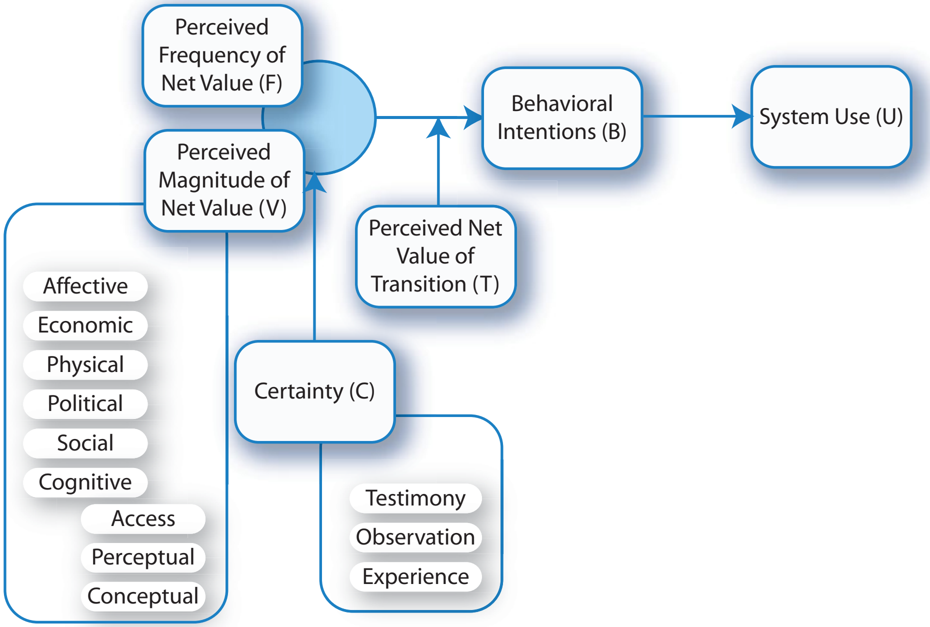


Figure 6.1. Technology transition model (TTM). TTM can be used for an evaluation of acceptance of use of implementations of ArcIMap in their context. Diagram after (Briggs et al., 2001).

- *Economic:* The extent to which the technology will increase or decrease the user's cash, assets, marketability, etc.
- *Physical:* The extent to which the technology will increase or decrease the user's health or comfort.
- *Political:* The extent to which the technology will increase or decrease the user's power or influence within and/or across organizations.
- *Social*: The extent to which the technology will enhance or detract from the user's personal relationships with other people, such as colleagues, friends, and family.
- *Cognitive*: The extent to which the technology will increase or decrease the user's amount of mental effort expended to complete tasks the technology supports. This dimension has at least three components:
  - *Perceptual load – user friendliness:* The amount of mental effort required finding and controlling the features and functions of the technology required to accomplish the task at hand.
  - *Access load – availability*: The amount of mental effort required to gain permission and access to the use of the components of the software needed for the task at hand.
  - *Conceptual load – understanding*: The amount of effort required to understand what the software is supposed to do for the user." (Agres et al., 2005: 271-272)

Figure 6.1 shows the technology transition model in its entirety.

The constructs of TTM can be measured by using structured interviews and questionnaires. These can be self developed or existing and well tested questionnaires can be utilized. TTM can already be applied at the design level of an application of ArcIMap, and followed through until the software usability evaluation stage.

BIBLIOGRAPHY

Achten, H. (1997). *Generic representations,* The Netherlands, Eindhoven University of Technology. Ph.D. dissertation.

Ackoff, R.L. (1989). From data to wisdom. *Journal of Applied Systems Analysis* 16**,** 3-9.

Agres, A.B., Vreede, G.-J.D. and Briggs, R.O. (2005). A tale of two cities: Case studies of group support systems transition. *Group Decision and Negotiation* 14**,** 267-284.

Akin, Ö. (1989). *Psychology of architectural design,* London, Pion Ltd.

Akin, Ö. (2001). Simon says: Design is representation. *Arredamento* July.

Akin, Ö. (2002). Case-based instruction strategies in architecture. *Design Studies* 23(4)**,** 407-431.

Akin, Ö. and Akin, C. (1996). Expertise and creativitiy in architectural design. IN Akin, Ö. and Saglamoglu, G. (Eds.) *Proceedings of the First International Symposium on Descriptive Models of Design.* Taskisla, Istanbul, Turkey.

Akin, O., Cumming, M., Shealey, M. and Tunçer, B. (1997). An electronic design assistance tool for case-based representa-tion of designs. *Automation in Construction* 6**,** 265-274.

Anderson, O.R. (1992). Some interrelationships between constructivist models of learning and current neurobiological theory, with implications for science education. *Journal of Research in Science Teaching* 19(10)**,** 1037-1058.

Archer, L.B. (1979). Whatever became of design methodology? *Design Studies* 1(1)**,** 17-20.

Argan, G.C. (1963). On the typology of architecture. *Architectural Design* December**,** 564-565.

Aroyo, L. (2001). *Task-oriented approach to information handling support within web-based education,* Enschede, University of Twente.

Asimow, M. (1962). *Introduction to design,* Englewood, Prentice Hall.

Ausubel, D.P. (1968). *Educational psychology: A cognitive view,* New York, Holt, Rinehart and Winston, Inc.

Axelrod, R. (Ed.) (1976). *The structure of decision,* Princeton, NJ, Princeton University Press.

Aygen, Z. (1998). *A hybrid model for case indexing and retrieval in building design,* Pittsburgh, PA, Carnegie Mellon University.

Bailey, K.D. (1994). *Typologies and taxonomies: An introduction to classification techniques,* Thousand oaks, CA, Sage.

Banathy, B.H. (1991). Cognitive mapping of educational systems for future generations. *World Future* 30(1)**,** 5-17.

Baron, J.B. and Steinberg, R. (Eds.) (1987). *Teaching thinking skills: Theories and practice,* New York, Freeman.

Barrow, H.G. and Tenenbaum, J.M. (1981). Computational vision. *Proceedings of the IEEE* 69(5)**,** 572-595.

Bates, M J. (1986). An exploratory paradigm for online information retrieval. IN Brooks, B.C. (Ed. *Intelligent Information Systems for the Information Society: Proceedings of the 6th IRFIS Conference.* New York, North-Holland.

Belkin, N.J. (2000). Helping people find what they don't know. *Communications of the ACM* 43(8)**,** 58-61.

Belkin, N.J. and Croft, W.B. (1987). Retrieval techniques. *Annual Review of Information Science and Technology* 22**,** 109-145.

Berger, A.A. (1995). *Cultural criticism: A primer of key concepts,* Thousand Oakes, CA, Sage Publications.

Bhatta, S.R., Goel, A.K. and Prabhakar, S. (1994). Innovation in analogical design: A model-based approach. IN Gero, J. and Sudweeks, F. (Eds.) *Artificial Intelligence in Design'94.* Kluwer Academic Publishers.

Biezunski, M., Bryan, M. and Newcomb, S. (Eds.) (2002). *Iso/iec 13250, topic maps (second edition)*.

Bishop, C.M. (1995). *Neural networks for pattern recognition,* Oxford, Clarendon.

Blaser, W. (1989). *Santiago calatrava: Engineering architecture,* Berlin, Birkhauser Verlag.

Bowker, G.C. and Star, S.L. (1999). *Sorting things out: Classification and its consequences,* Cambridge, Mass. and London, England, MIT Press.

Briggs, R., Nunamaker, J. and Tobey, D. (2001). The technology transition model: A key to self-sustaining and growing communities of gss users. *34th annual hawaii international conference on system sciences (hicss-34).* 1061-1069.

Briggs, R.O., Adkins, M., Mittleman, D., Kruse, J., Miller, S. and Nunamaker, J.F., Jr. (1999). A technology transition model derived from field investigation of gss use aboard the u.S.S. Coronado. *Journal of Management Information Systems* 15(3)**,** 151-195.

Broadbent, G. (1988). *Design in architecture: Architecture and the human sciences,* London, UK, David Fulton Publishers.

Buzan, T. and Buzan, B. (1996). *The mind map book: How to use radiant thinking to maximize your brain's untapped potential,* New York, Penguin.

Carrara, G., Kalay, Y.E. and Novembri, G. (1992). Multi-modal representation of design knowledge. *Automation in Construction* 1(2)**,** 111-121.

Casakin, H. (1997). *The role of analogy and visual displays in architectural design,* Haifa, Israel, Technion-Israel Institute of Technology.

Casakin, H. (2004). Visual analogy as a cognitive strategy in the design process: Expert versus novice performance. *Journal of Design Research* 4(2).

Casakin, H. and Dai, W. (2002). Visual typology in design: A computational view. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM)* 16(1)**,** 3-21.

Casakin, H. and Goldschmidt, G. (1999). Expertise and the use of visual analogy: Implications for design education. *Design Studies* 20(2)**,** 153-175.

Casakin, H. and Goldschmidt, G. (2000). Reasoning by visual analogy in design problem-solving: The role of guidance. *Journal of Planning and Design: Environment & Planning B* 27(1)**,** 105-119.

Chase, S.C. (1989). Shapes and shape grammars: From mathematical model to computer implementation. *Environment and Planning B: Planning and Design* 16**,** 215-242.

Chaszar, A. (Ed.) (2006). *Blurring the lines: Computer-aided-design and -manufacturing in architecture,* London, Wiley-Academy.

Ching, F.D.K. (1979). *Architecture: Form, space and order,* New York, Van Nostrand Reinhold.

Chiu, M.-L. and Shih, S.-G. (1997). Analogical reasoning and case adaptation in architectural design: Computers vs. Human designers. *Proceedings of CAAD Futures 1997.* München, Germany.

Chu, S. and Cesnik, B. (2001). Knowledge representation and retrieval using conceptual graphs and free text document self-organisation techniques. *International Journal of Medical Informatics* 62(2-3)**,** 121-133.

Churchman, C.W. (1967). Wicked problems. *Management Science* 14(4)**,** B141-142.

Çiftçioglu, Ö., Durmisevic, S., Durmisevic, E. and Sariyildiz, I.S. (1999). Artificial intelligence in building design. IN Friedman, N.A. and Barrallo, J. (Eds.) *Isama 99.* San Sebastian, The University of the Basque Country**,** 113-120.

Corbett, D. (2003). *Reasoning and unification over conceptual graphs,* New York, Kluwer Academic.

Corbett, D.R. and Burrow, A.L. (1996). Knowledge reuse in seed exploiting conceptual graphs. *Proceedings Fourth International Conference on Conceptual Structures.* Sydney, Australia, UNSW Press.

Croft, W. and Cruse, D A. (2004). *Cognitive linguistics,* Cambridge, UK, Cambridge University Press.

Cross, N. (1982). Designerly ways of knowing. *Design Studies* 3(4)**,** 221-227.

Cross, N. (2004). Expertise in design: An overview. *Design Studies* 25(5)**,** 427-441.

Cross, N., Christiaans, H. and Dorst, K. (1996). Introduction: The delft protocols workshop. IN Cross, N., Christiaans, H. and Dorst, K. (Eds.) *Analysing design activity.* Chichester, UK, John Wiley & Sons.

Cross, N. and Cross, A.C. (1998). Expert designers. IN Frankenberger, E., Badke-Schaub, P. and Birkhofer, H. (Eds.) *Designers: The key to successful product development.* London, Springer.

Cruse, D.A. (1986). *Lexical semantics,* Cambridge, Cambridge University Press.

Cruse, D.A. (2002). Hyponomy and its varieties. IN Green, R., Bean, C.A. and Myaeng, S.H. (Eds.) *The semantics of relationships: An interdisciplinary perspective.* Dordrecht, Kluwer Academic Publishers**,** 3-21.

Cumming, M. (2005). *Constructing process models from distributed design activity,* Pittsburgh, PA, Carnegie Mellon University. dissertation.

Dave, B., Schmitt, G., Faltings, B. and Smith, I. (1994). Case-based design in architecture. IN Gero, J.S. and Sudweeks, F. (Eds.) *Artificial Intelligence in Design '94.* Dordrecht (The Netherlands), Kluwer Academic.

Davis, F.D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *Management Information Systems Quarterly* 13(3)**,** 319-340.

Davis, J.G., Subrahmanian, E., Konda, S., Granger, H., Collins, M. and Westerberg, A.W. (2001). Creating shared information spaces to support collaborative design work. *Information Systems Frontiers* 3(3)**,** 377-392.

de Bono, E. (1994). *Parallel thinking,* London, Penguin Books Ltd.

de Jong, T.M. and Voordt, T.J.M.v.d. (2000). Retrieval and reference. IN Jong, T.M.d., Cuperus, Y.J. and Voordt, T.J.M.v.d. (Eds.) *Ways to study: Architectural, urban, and technical design.* Delft, Delft University of Technology**,** 29-35.

De Vries, M. and Wagter, H. (1990). A caad model for use in early design phases. IN McCullough, M., Mitchell, W.J. and Purcell, P. (Eds.) *The electronic design studio: Architectural knowledge and media in the computer era.* Cambridge, MA, MIT Press**,** 215-228.

Dichev, C., Dicheva, D. and Aroyo, L. (2003). Using topic maps for e-learning. *Computers and advanced technology in education (cate 2003).* Rhodes, Greece**,** 26-31.

Domeshek, E.A. and Kolodner, J.L. (1992). A case-based design aid for architecture. IN Gero, J S. (Ed. *Artificial Intelligence in Design '92.* Dordrecht (The Netherlands), Kluwer Academic.

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

Dori, D. (2002). *Object-process methodology,* Berlin, Springer-Verlag.

Dorst, K. (1997). *Describing design. A comparison of paradigms,* Delft, The Netherlands, Delft University of Technology.

Eastman, C.M. (1999). *Building product models: Computer environments supporting design and construction,* Boca Raton, FL, RC Press, Inc.

Eckert, C. and Stacey, M. (2000). Sources of inspiration: A language of design. *Design Studies* 21(5)**,** 523-538.

Eden, C., Jones, S. and Sims, D. (1979). *Thinking in organizations,* London, Macmillan.

Egli, H.G. (1997). *Sinan: An interpretation,* Istanbul, Ege Yayinlari.

Engeli, M. (Ed.) (2001). *Bits and spaces,* Basel, Birkhäuser.

Ertug, A. (1981). Concept of form and space in ottoman architecture. IN Ertug, A. (Ed.) *Space and tradition: Turkish architecture.* Tokyo Process Architecture Publishing.

Erzen, J.N. (1996). *Mimar sinan: Estetik bir analiz,* Ankara, Sevki Vanli Mimarlik Vakfi Yayinlari.

Falkenhainer, B., Forbus, K.D. and Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence* 41(1)**,** 1-63.

Fang, N. (1993). *A knowledge-based computational approach to architectural precedent analysis,* Delft, The Netherlands, Delft University of Technology.

Fellbaum, C. (Ed.) (1998). *Wordnet: An electronic lexical database,* Cambridge, MA, MIT Press.

Flemming, U. (1990). Syntactic structures in architecture: Teaching composition with computer assistance. IN McCullough, M., Mitchell, W.J. and Purcell, P. (Eds.) *The electronic design studio.* Cambridge, Mass., The MIT Press**,** 31-48.

Flemming, U. and Woodbury, R. (1995). Software environment to support early phases in building design: Overview. *Journal of Architectural Engineering* 4(1)**,** 147-152.

Ford, K.M., Coffey, J.W., Cañas, A.J., Andrews, E.J. and Turner, C.W. (1996). Diagnosis and explanation by a nuclear cardiology expert system. *International Journal of Expert Systems* 9**,** 499-506.

Fraser, K.M. (1993). *Theory based use of concept mapping in organization development: Creating shared understanding as a basis for the cooperative design of work changes and changes in working relationships,* Ithaca, NY, Cornell University.

Frederiksson, M. and Gustavsson, R. (2001). A methodological perspective on engineering of agent societies. IN Omicini, A., Petta, P. and Tolkdorf, R. (Eds.) *2nd international workshop on emerging societies in the agent's world (esaw'01).* Berlin Heidelberg, Springer-Verlag.

Freese, E., Ahmed, K., Park, J. and Hunting, S. (2003). Open source topic map software. IN Park, J. and Hunting, S. (Eds.) *Xml topic maps: Creating and using topic maps for the web.* Boston, MA, Addison-Wesley.

Gaines, B.R. and Shaw, M.L.G. (1992). Integrated knowledge acquisition architectures. *Journal for Intelligent Information Systems* 1(1)**,** 9-34.

Gaines, B.R. and Shaw, M.L.G. (1995). Collaboration through concept maps. *Proceedings of cscl95: Computer supported cooperative learning.* Bloomington.

Garshol, L.M. (2004). Research in action: Metadata? Thesauri? Taxonomies? Topic maps! Making sense of it all. *Journal of Information Science* 30(4)**,** 378-391.

Gero, J.S. (1990). Design prototypes: A knowledge representation schema for design. *AI Magazine* 11(4)**,** 26-36.

Gero, J.S. and Maher, M.L. (1992). Mutation and analogy to support creativity in computer aided design. IN Schmitt, G. (Ed. *CAAD Futures `91.* Wiesbaden, Wieweg.

Glaser, B.G. and Strauss, A.L. (1967). *The discovery of grounded theory, strategies for qualitative research,* Chicago, Aldine Publishers.

Goldschmidt, G. (1994). Visual analogy in design. IN Trappl, R. (Ed.) *Cybernetics and systems '94.* Singapore, World Scientific**,** 507-514.

Goldschmidt, G. (1995). Visual displays for design: Imagery, analogy and databases of visual images. IN Koutamanis, A., Timmermans, H. and Vermeulen, I. (Eds.) *Visual databases in architecture.* Aldershot, UK, Avebury.

Goodwin, G. (1971). *A history of ottoman architecture,* New York, Thames and Hudson.

Graesser, A.C. and Clark, L.F. (1985). *Structures and procedures of implicit knowledge,* New Jersey, Ablex.

Grand, B.L. and Soto, M. (2003). Topic maps visualization. IN Geroimenko, V. and Chen, C. (Eds.) *Visualizing the semantic web: Xml-based internet and information visualization.* London, Springer Verlag**,** 49-62.

Green, R., Bean, C.A. and Myaeng, S.H. (Eds.) (2002). *The semantics of relationships: An interdisciplinary perspective,* Dordrecht, Kluwer Academic Publishers.

Greenberg, I. (1999a). Facing up to new interfaces. *IEEE Computer* 32(4)**,** 14-16.

Greenberg, I. (1999b). Facing up to new interfaces. *IEEE Computer* 4(32)**,** 14-16.

Groen, P., Kersten, A. and de Zeeuw, G. (1980). *Beter sociaal veranderen,* Muiderberg, Netherlands, Coutinho.

Gross, M. (1995). Indexing visual databases of designs with diagrams. IN Koutamanis, A., Timmermans, H. and Vermeulen, I. (Eds.) *Visual databases in architecture.* Aldershot, UK, Avebury**,** 1-14.

Gross, M. and Do, E. (1995). Drawing analogies - supporting creative architectural design with visual references. IN Maher, M.-L. and Gero, J. (Eds.) *3d International Conference on Computational Models of Creative Design.* Sydney: University of Sydney.

Gruber, T.R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2)**,** 199-220.

Henninger, S. (1997). An evolutionary approach to constructing effective software reuse repositories. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 6(2)**,** 111-140.

Herschel, R.T., Nemati, H. and Steiger, D. (2001). Tacit to explicit knowledge conversion: Knowledge exchange protocols. *Journal of Knowledge Management* 5(1)**,** 107-116.

Heylighen, A. (2000). *In case of architectural design: Critique and praise of case-based design in architecture,* Leuven, Belgium, K.U. Leuven. Ph.D. dissertation.

Heylighen, A. and Neuckermans, H. (2000). Dynamo - dynamic architectural memory on-line. *Journal of Educational Technology and Society* 3(2)**,** 86-95.

Humphreys, G.W. and Bruce, V. (1989). *Visual cognition,* London, Lawrence Erlbaum Associates.

Inoue, K. and Urahama, K. (2000). Learning of view-invariant pattern recognizer with temporal context. *Pattern Recognition* 33**,** 1665-1674.

Jacobson, I., Christerson, M., Jonsson, P. and Overgaard, G. (1992). *Object-oriented software engineering: A use case driven approach,* Reading, MA, Addison-Wesley.

Jakobson, R. (1999). The metaphoric and metonymic poles. IN Lodge, D. and Wood, N. (Eds.) *Modern criticism and theory: A reader.* New York, Longman.

Johnson, D.W. and Johnson, R.T. (1991). *Learning together and alone: Cooperative, competitive, and individualistic learning,* Englewood Cliffs, NJ, Prentice-Hall.

Jokinen, T. and Maurer, B. (Eds.) (1998). *"Der magus des nordens": Alvar aalto und die schweiz,* Zurich, GTA.

Kaplan, S., Gruppen, L., Leventhal, L.M. and Board, F. (1986). *The components of expertise: A cross-disciplinary review,* Ann Arbor, The University of Michigan.

Keller, I. (2005). *For inspiration only: Designer interaction with informal collections of visual material,* Delft, The Netherlands, Delft University of Technology.

Kocatürk, T. and Tunçer, B. (2004). Modeling collaborative conceptual design in the digital age. IN Dikbas, A. (Ed.) *Third international workshop on construction information technology in education.* Istanbul, Turkey, ITU Press**,** 149-163.

Kocaturk, T., Veltkamp, M. and Tunçer, B. (2003). Exploration of interrelationships between digital design and production processes of free-form complex surfaces in a web-based database. IN Chiu, M.-L., Tsou, J.-Y., Kvan, T., Morozumi, M. and Jeng, T.-S. (Eds.) *Digital design: Research and practice, proceedings of the 10th international conference on computer aided architectural design futures.* Dordrecht, The Netherlands, Kluwer Academic**,** 445-454.

Kolodner, J. (1993). *Case-based reasoning,* San Mateo CA, Morgan Kaufmann.

Kooistra, J. (2002). Flowing. *Systems Research and Behavioral Science* 19**,** 123-127.

Kooistra, J. and Hopstaken, C.W.J. (2002). Ommat: Dealing with electronic scientific information. IN Brophy, P. (Ed.) *Libraries without walls 4; the delivery of library services to distant users.* London Facet.

Kooistra, J., Hopstaken, C.W.J., Ertsen, M. and Lagerwei, N.J. (2004). Supporting information-based networks in higher education. IN Brophy, P., Fisher, S. and Craven, J. (Eds.) *Libraries without walls 5: The distributed delivery of library and information services.* London, Facet publishing.

Kooistra, J., Hopstaken, C.W.J., Stouffs, R., Tunçer, B., Janssen Groesbeek, E. and Sjoer, E. (2005). Keyset: Conceptual, technical, didactic and strategic qualities of a 'discourse browser'. IN Sariyildiz, S. and Tunçer, B. (Eds.) *Proceedings of the 3rd international conference on innovation in architecture, engineering and construction.* Rotterdam, SOON**,** 353-364.

Kooistra, J., Stouffs, R. and Tunçer, B. (2003). Metadata as a means for correspondence in design analysis. IN Tunçer, B., Ozsariyildiz, S S. and Sariyildiz, S. (Eds.) *E-Activities in Design and Design Education.* Paris, Europia.

Koutamanis, A. (1995). Recognition and retrieval in visual architectural databases. IN Koutamanis, A., Timmermans, H. and Vermeulen, I. (Eds.) *Visual databases in architecture.* Aldershot, UK, Avebury.

Kremer, R. (1997). *Constraint graphs: A concept map meta-language,* CALGARY, The University of Calgary.

Krishnamurti, R. (1981). The construction of shapes. *Environment and Planning B: Planning and Design* 8**,** 5-40.

Krishnamurti, R. and Earl, C.F. (1992). Shape recognition in three dimensions. *Environment and Planning B: Planning and Design* 19**,** 585-603.

Krishnamurti, R. and Stouffs, R. (1997). Spatial change: Continuity, reversibility and emergent shapes. *Environment and Planning B: Planning and Design* (24)**,** 359-384.

Kuban, D. (1980). Architecture of the ottoman period. IN Akurgal, E. (Ed.) *The art and architecture of turkey.* Fribourg, Switzerland, Office du Livre.

Kuban, D. (1987). The style of sinan's domed structures. IN Grabar, O. (Ed.) *Muqarnas; an annual on islamic art and architecture.* Leiden, Brill.

Kumar, A. and Kahle, D.J. (2006). Vue: A concept mapping tool for digital content. IN Cañas, A.J. and Novak, J.D. (Eds.) *Concept maps: Theory, methodology, technology, proc. Of the second international conference on concept mapping.* San José, Costa Rica.

Lambiotte, J.G., Dansereau, D.F., Cross, D.R. and Reynolds, S.B. (1989). Multirelational semantic maps. *Educational Psychology Review* 1(4)**,** 331-367.

Lang, J. (1987). *Creating architectural theory: The role of the behavioral sciences in environmental design,* New York, Van Nostrand Reinhold.

Lave, J. and Wenger, E. (1991). *Situated learning: Legitimate peripheral participation,* Cambridge, University of Cambridge Press.

Lawson, B. (2004). Schemata, gambits and precedent: Some factors in design expertise. *Design Studies* 25(5)**,** 443-457.

Lawson, B. (2005). Oracles, draughtsmen, and agents: The nature of knowledge and creativity in design and the role of it. *Automation in Construction* 14(3)**,** 383-391.

Lawson, B. and Loke, S.M. (1997). Computers, words and pictures. *Design Studies* 18(2)**,** 171-183.

Lee, J. (2005). Model-driven business transformation and the semantic web. *Communications of the ACM* 48(12)**,** 75-77.

Leupen, B., Grafe, C., Körnig, N., Lampe, M. and de Zeeuw, P. (1997). *Design and analysis,* Rotterdam, 010.

Lin, X. (1995). Searching and browsing on map displays. *Proceedings of asis'95.* Chicago**,** 13-18.

Lin, X. (1997). Map displays for information retrieval. *Journal of the American Society for Information Science* 48(1)**,** 40-54.

Linnaeus, C. (1964 [1735]). *Systema naturae,* New York, Stechert-Hafner.

Lloyd, P. (2000). Storytelling and the development of discourse in the engineering design process. *Design Studies* 21(4)**,** 357-373.

Madrazo, L. (1995). *The concept of type in architecture: An inquiry into the nature of architectural form,* Zurich, Switzerland, Swiss Federal Institute of Technology. Ph.D. dissertation.

Madrazo, L. (1999). Types and instances: A paradigm for teaching design with computers. *Design Studies* 20(2)**,** 177-193.

Madrazo, L. and Weder, A. (2001). Aalto on the internet: Architectural analysis and concept representation with computer media. *Automation in Construction* 10(5)**,** 561-575.

Maher, M.L., Balachandran, M.B. and Zhang, D.M. (1995). *Case-based reasoning in design,* Mahwah, NJ, Lawrence Erlbaum.

Marchionini, G. (1995). *Information seeking in electronic environments,* New York, Cambridge University Press.

Martin, P. and Eklund, P. (1999). Embedding knowledge in web documents. *Computer Networks* 33(11-16)**,** 1403-1419.

Martin, P.Y. and Turner, B.A. (1986). Grounded theory and organizational research. *Journal of Applied Behavioral Science* 22(2)**,** 141-157.

McDonnell, J., Lloyd, P. and Valkenburg, R.C. (2004). Developing design expertise through the construction of video stories. *Design Studies* 25(5)**,** 509-525.

Megill, K.A. and Schantz, H.F. (1999). *Document management: New technologies for the information services manager,* London, Bowker-Saur.

Miller, G.A. (1995). Wordnet: A lexical database for english. *Communications of the ACM* 38(11)**,** 39-41.

Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K.J. (1990). Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography* 3(4)**,** 235-244.

Minsky, M. (1975). A framework for representing knowledge. IN Winston, P. (Ed.) *The psychology of computer vision.* New York, McGraw-Hill**,** 211-277.

Mitchell, W.J. (1990). *The logic of architecture: Design, computation, and cognition,* Cambridge, Mass., The MIT Press.

Mitchell, W.J. (1994). Three paradigms for computer-aided design. *Automation in Construction* 3**,** 239-245.

Moneo, R. (1978). On typology. *Oppositions* 13**,** 23-45.

Moore, G., Bogachev, D. and Nishikawa, M. (Eds.) (2005). *Iso n458: Topic maps constraint language*.

Motro, A. (1986). Baroque: A browser for relational databases. *ACM Transactions on Office Information Systems* 4**,** 164-181.

Mubarak, K. (2004). *Case based reasoning for design composition in architecture,* Pittsburgh, PA, Carnegie Mellon University.

Muller, W. and Pasman, G. (1996). Typology and the organization of design knowledge. *Design Studies* 17(2)**,** 111-130.

Murphy, M.L. (2003). *Semantic relations and the lexicon*, Cambridge University Press.

Murthy, H. and Lutton, L. (1994). Interactive multimedia based intelligent systems in architecture. IN McBeath, C. and Atkinson, R. (Eds.) *Proceedings of the Second International Interactive Multimedia Symposium.* Perth, Western Australia, Promaco.

Murty, P. and Purcell, T. (2002). Discovery processes in designing. *Proceedings of ANZAAScA2002.* School of Architecture + Building, Deakin University.

Murty, P. and Purcell, T. (2003). Unexpected discovery processes in designing. *ANZAAScA Conference.* Sydney.

Newell, A. and Simon, H. (1972). *Human problem solving,* New York, NY, Prentice-Hall Inc.

Nonaka, I. and Takeuchi, H. (1995). *The knowledge-creating company: How japanese companies create the dynamics of innovation,* New York, Oxford University Press.

Novak, J.D. (1998). *Learning, creating, and using knowledge: Concept maps as facilitative tools in schools and corporations,* Mahwah, NJ, Lawrence Erlbaum Associates.

Novak, J.D. (2002). The theory underlying concept maps and how to construct them.

Novak, J.D. and Gowin, D.B. (1984). *Learning how to learn,* Cambridge, UK, Cambridge University Press.

Ortony, A. (Ed.) (1993). *Metaphor and thought,* Cambridge, Cambridge University Press.

Oxman, R. (2004). Think-maps: Teaching design thinking in design education. *Design Studies* 25(1)**,** 63-91.

Oxman, R. and Oxman, R. (1993). Precedents: Memory structure in design case libraries. IN Flemming, U. and Wyk, S.V. (Eds.) *Proceedings of the Fifth international Conference on Computer-Aided Architectural Design Futures.* Pittsburgh, Pennsylvania, United States, North-Holland Publishing Co.,Amsterdam, The Netherlands.

Oxman, R. and Oxman, R. (1994). Remembrance of things in the past: Design precedents in libraries. IN Tzonis, A. and White, I. (Eds.) *Automation based creative design.* Amsterdam, Elsevier Science**,** 55-68.

Oxman, R.E. (1994a). A computational model for the organization of case knowledge of a design precedent. *Design Studies* 15(2)**,** 141-157.

Oxman, R.E. (1994b). Precedents in design: A computational model for the organization of precedent knowledge. *Design Studies* 15(2)**,** 141-157.

Papanikolaou, M. and Tunçer, B. (1999). The fake.Space experience - exploring new spaces. IN Brown, A., Knight, M. and Berridge, P. (Eds.) *Architectural computing: From turing to 2000.* Liverpool, UK, eCAADe and The University of Liverpool**,** 395-402.

Papanokolaou, M. (2001). It in virtual enterprises. IN Engeli, M. (Ed.) *Bits and spaces: Architecture and computing for physical, virtual, hybrid realms.* Basel, Birkhäuser**,** 172-175.

Park, J. and Hunting, S. (2003). *Xml topic maps: Creating and using topic maps for the web,* Boston, MA, Addison-Wesley.

Pasman, G. (2003). *Designing with precedents,* Delft, The Netherlands, Delft University of Technology.

Pepper, S. (2002). The tao of topic maps: Finding the way in the age of infoglut. Ontopia.

Pepper, S. and Moore, G. (Eds.) (2001). *Xml topic maps (xtm) 1.0*.

Pidcock, W. (2003). What are the differences between a vocabulary, a taxonomy, a thesaurus, an ontology, and a meta-model?

Plumb Design (1998). Visual thesaurus.

Polanyi, M. (1966). *The tacit dimension,* Garden City, Doubleday.

Popovic, V. (2002). General strategic knowledge models and their interaction with domain-specific knowledge in design. IN Durling, D. and Shackleton, J. (Eds.) *Common Grounds Proceedings.* Staffordshire, Staffordshire University Press.

Popper, K.R. (1982). *The open universe: An argument for indeterminism,* London, Hutchinson.

Pribbenow, S. (2002). Meronymic relationships: From classical mereology to complex part-whole relations. IN Green, R., Bean, C A. and Myaeng, S.H. (Eds.) *The semantics of relationships: An interdisciplinary perspective.* Dordrecht, Kluwer Academic Publishers**,** 35-50.

Prieto-Díaz, R. (1991). Implementing faceted classification for software reuse. *Communications of the ACM* 34(5)**,** 88-97.

Prigogine, I. and Stengers, I. (1984). *Order out of chaos,* New York, Bantam Books.

Qian, L. and Gero, J.S. (1992). A design support system using analogy. IN Gero, J.S. (Ed. *Artificial Intelligence in Design '92.* Kluwer, Dordrecht.

Quillian, M.R. (1968). Semantic memory. IN Minsky, M. (Ed.) *Semantic information processing.* Cambridge, Mass., The MIT Press.

Ranganathan, S.R. (1962). *Elements of library classification,* Bombay, Asia Publishing House.

Reed, P. (Ed.) (1998). *Alvar aalto: Between humanism and materialism,* New York, The Museum of Modern Art.

Restrepo, J. (2004). *Information processing in design,* Delft, The Netherlands, Delft University of Technology.

Rittel, H. (1973). The state of the art in design methods. *Design Research and Methods (Design Methods and Theories)* 7(2)**,** 143-147.

Rossi, A. (1985). Ten opinions on typology. *Casabella* 49**,** 509-510.

Rowe, P. (1987). *Design thinking,* Cambridge, MA, MIT Press.

Rumbaugh, J., Jacobson, I. and Booch, G. (2005). *The unified modeling language reference manual, second edition,* Reading, Addison-Wesley.

Saaty, T.L. (1980). *The analytic hierarchy process,* New York, N.Y., McGraw Hill.

Sariyildiz, I S., Stouffs, R. and Çiftçioglu, Ö. (2002). Intelligent modeling for cooperative engineering. IN Turk, Z. and Scherer, R. (Eds.) *eWork and eBusiness in Architecture, Engineering and Construction.* Lisse, The Netherlands, A.A. Balkema.

Sariyildiz, S. (1991). *Conceptual design by means of islamic-geometric-patterns within a caad-environment,* Delft, Netherlands, TU Delft. PhD Dissertation.

Schmitt, G. (1993). *Architectura et machina: Computer aided architectural design und virtuelle architektur,* Braunschweig, Germany, Vieweg.

Schmitt, G. (1994). Case-based design and creativity. IN A., T. and White, I. (Eds.) *Automation based creative design.* Netherlands, Elsevier**,** 41-53.

Schön, D A. (1983). *The reflective practitioner: How professionals think in action,* New York, Basic Books.

Schön, D A. (1985). *The design studio: An exploration of its traditions & potential,* London, RIBA Publications.

Schön, D A. (1988). Designing: Rules, types and words. *Design Studies* 9(3)**,** 181-190.

Schön, D A. (1993). Generative metaphor: A perspective on problem-setting in social policy. IN Ortony, A. (Ed.) *Metaphor and thought.* Cambridge, UK, Cambridge University Press**,** 137-163.

Simon, H. (1969). *The sciences of the artificial,* Cambridge, MA, MIT Press.

Simon, H. (1973). The structure of ill structured problems. *Artificial Intelligence* 4(3)**,** 181-201.

Simon, H.A. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society* 106**,** 467-482.

Smith, I., Lottaz, C. and Faltings, B. (1995). Spatial composition using cases: Idiom. *Cbr research and development, lecture notes in ai 1010.* Berlin, Springer-Verlag**,** 88-97.

Smith, S. (2000). Team up on the internet. *CADALYST* 17(February)**,** 52-58.

Sowa, J.F. (1983). Generating language from conceptual graphs. *Comp. & Math. with Applic.* 9(1)**,** 29-44.

Sowa, J.F. (1984). *Conceptual structures: Information processing in mind and machine,* Reading, MA, Addison-Wesley.

Sowa, J.F. (Ed.) (1991). *Principles of semantic networks: Explorations in the representation of knowledge,* San Mateo, CA, Morgan Kaufmann Publishers.

Sowa, J.F. (2000). *Knowledge representation: Logical, philosophical, and computational foundations,* Pacific Grove, CA, Brooks Cole Publishing Co.

Sowa, J.F. (2006). Semantic networks.

Spence, R. (2001). *Information visualization,* Boston, MA, Addison-Wesley.

Steadman, P. (1979). *The evolution of designs: Biological analogy in architecture and the applied arts,* Cambridge, UK, Cambridge University Press.

Steenbergen, C., Mihl, H., Reh, W. and Aerts, F. (Eds.) (2002). *Architectural design and composition,* Bussum/Delft, THOTH publishers & Faculty of Architecture TU Delft.

Stierlin, H. (1985). *Soliman et l'architecture ottomane,* Fribourg, Switzerland, Office du Livre.

Stierlin, H. (1998). *Turkey: From the selçuks to the ottomans,* Cologne, Taschen.

Stouffs, R. (2001). Visualizing information structures and its impact on project teams: An information architecture for the virtual aec company. *Building Research & Information* 29(3)**,** 218-232.

Stouffs, R., Kooistra, J. and Tunçer, B. (2004a). Metadata as a means for correspondence on digital media. *ITcon* 9(Special Issue Digital Media Libraries)**,** 129-142.

Stouffs, R., Kurmann, D., Tunçer, B., Mieusset, K.H. and Stäger, B. (1998). An information architecture for the virtual aec company. IN Amor, R. (Ed. *Product and Process Modelling in the Building Industry.* Watford, UK, Building Research Establishment.

Stouffs, R., Tunçer, B., Janssen Groesbeek, E. and Kooistra, J. (2004b). Keyset: The use of metadata as a means to improve the quality of design. IN Dikbas, A. (Ed.) *Third international workshop on construction information technology in education.* Istanbul, Turkey, ITU Press**,** 79-94.

Stouffs, R., Tunçer, B., Janssen Groesbeek, E., Kooistra, J. and Sjoer, E. (2005). The use of metadata as a means to improve the quality of design: Results from education. IN Bhatt, A. (Ed.) *Caadria 2005 (proceedings of the 10th international conference on computer aided architectural design research in asia).* New Delhi, India, TVB School of Habitat Studies**,** 503-509.

Stouffs, R., Tunçer, B., Venne, R.F. and I.S.Sariyildiz (2002). Infobase: A multimedia learning environment to support group work. IN Rebolj, D. (Ed.) *Construction information technology in education.* Rotterdam, International Council for Research and Innovation in Building and Construction**,** 75-82.

Stouffs, R. and Wieringa, M. (2006). The generation of chinese ice-ray lattice structures for 3d façade design. IN Dulaimi, M. (Ed. *Conference Proceedings of The Joint International Conference on Construction Culture, Innovation and Management (CCIM).* Dubai, UAE, BUiD.

Todras-Whitehill, E. (2005). 'folksonomy' carries classifieds beyond swf and 'for sale'. *The New York Times.*

Tolman, F.P. (1999). Product modeling standards for the building and construction industry: Past, present and future. *Automation in Construction* 8**,** 227-235.

Tunçer, B. (2007). Modeling of knowledge for free form building processes. IN Dashti, H. (Ed.) *Sources of architectural form: Theory and practice.* Kuwait, Kuwait University.

Tunçer, B., Ciftcioglu, Ö., Sariyildiz, S. and Cumming, M. (2005). Intelligent design support. IN Bhatt, A. (Ed.) *Caadria 2005 (proceedings of the 10th international conference on computer aided architectural design research in asia).* New Delhi, India, TVB School of Habitat Studies**,** 436-446.

Tunçer, B. and Stouffs, R. (1999). Computational richness in the representation of architectural languages. IN Brown, A., Knight, M. and Berridge, P. (Eds.) *Architectural computing: From turing to 2000.* Liverpool, UK, eCAADe and The University of Liverpool**,** 603-610.

Tunçer, B., Stouffs, R. and Sariyildiz, I.S. (2002a). Document decomposition by content as a means for structuring building project information. *Construction Innovation* 2(4)**,** 229-248.

Tunçer, B., Stouffs, R. and Sariyildiz, I.S. (2002b). A toolkit for modelling architectural analyses in a design studio context. IN Rebolj, D. (Ed.) *Construction information technology in education.* Rotterdam, The Netherlands, International Council for Research and Innovation in Building and Construction**,** 59-64.

Turner, B.A. (1983). The use of grounded theory for the qualitative analysis of organisational behaviour. *Journal of Management Studies* 20(3)**,** 333-348.

Turner, S. and Turner, P. (2003). Telling tales: Understanding the role of narrative in the design of taxonomic software. *Design Studies* 24(6)**,** 537-547.

Tzitzikas, Y., Analyti, A., Spyratos, N. and Constantopoulos, P. (2004). An algebraic approach for specifying compound terms in faceted taxonomies. IN Kiyoki, Y., Kawaguchi, E., Jaakkola, H. and Kangassalo, H. (Eds.) *Information modelling and knowledge bases xv.* IOS Press**,** 67-87.

Tzonis, A. (1990). Huts, ships and bottleracks: Design by analogy for architects and/or machines. IN Cross, N., Dorst, K. and Roozenburg, N. (Eds.) *Research in design thinking.* Delft, NL, Delft University Press.

Tzonis, A. (1999). *Santiago calatrava - the poetics of movement,* London, Thames and Hudson.

Tzonis, A. (2004). *Santiago calatrava: The complete works,* New York, Rizzoli.

Tzonis, A. and White, I. (1994). *Automation based creative design: Research and perspectives,* Amsterdam, Elsevier.

van Leusen, M. (1994). *A system of types in the domain of residential buildings,* The Netherlands, Delft University of Technology. Ph.D. dissertation.

Vidler, A. (1977). The third typology. *Oppositions* 7**,** 1-4.

Way, E.C. (1991). *Knowledge representation and metaphor,* Dordrecht, Kluwer.

Way, E.C. (1994). Conceptual graphs - past, present, and future. IN Tepfenhart, W.M., Dick, J.P. and Sowa, J.F. (Eds.) *Proceedings of the 2nd international conference on conceptual structures : Current practices, lecture notes in artificial intelligence, volume 835.* Berlin, Springer**,** 11-30.

Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity,* London, Cambridge University Press.

Woo, J.-H., Clayton, M.J., Johnson, R.E., Flores, B.E. and Christopher, E. (2002). Dynamic knowledge map: Reusing experts' tacit knowledge in the aec industry. *Thresholds - Design, Research, Education and Practice, in the Space Between the Physical and the Virtual.* Pomona, ACADIA.

Yang, A. and Shan, Y. (Eds.) (2008). *Intelligent complex adaptive systems,* London, IGI Global.

APPENDICES

A

DESIGN ANALYSIS NETWORK SECOND YEAR DESIGN STUDIO ASSIGNMENT DESCRIPTION

This appendix contains the assignment description that was given to the students of the second year (4^{th} semester) design studio, where they used Design Analysis Network (DAN) for performing an architectural analysis. The assignment description also contains a manual of how to use the tool.

Workshop 1

Design analysis using DAN (Design Analysis Network) – InfoBase

In this workshop, we will do an analysis of a selected building using supplied documentation about this building. All the analysis results will be collected in a database such that the results of other students will be visible through the internet environment that we are using. The tools that are provided for creating the analyses also serve to relate the individual documents and add metadata to these. The result of this exercise will be an extensible library of precedent analyses. This work serves as a case study for the Ph.D. work of Bige Tunçer.

Use the links below to do the assignment:

- question 1
- search page
- question 2

Step by step explanation of the process:

Part 1:

- Create a group
- Choose an existing building to analyze from the electronic plannanmap
- Pick 2 planmiddelen that are important for the building you have chosen
- Analyze the building according to these using Design Analysis Network (DAN)
- Practice how to search the database using the metadata

Part 2:

- Write a short text (250 words) describing your design idea for your own design
- Choose two analyses as references to your design concept
- Print your analyses your design concept text and the two references and show these to your design teacher during critiques

How to work from home:

If you are having any problems with the analysis, please contact *Bige Tunçer* kab. 5.08a.

Step by step explanation of the process:

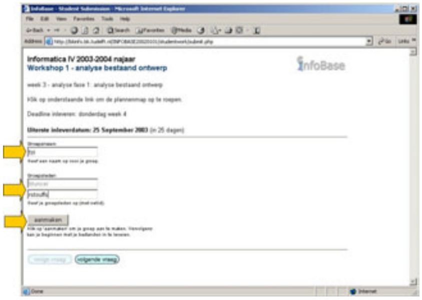
Part 1:

Create a group

This workshop will be preferably done in groups of two. If you want to analyze a building alone, you cal also work alone.

Go to question 1. You will first get the login window. NetID is your Blackboard username.

Once you log in, you will go to the webpage of question 1. Here, you first need to fill out the form to create a group.



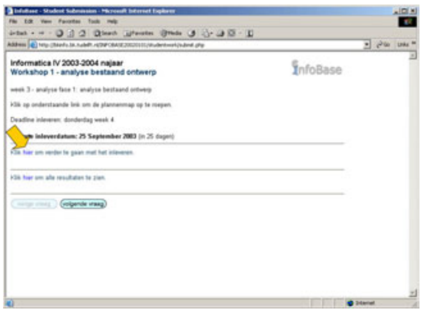
One person from each group should fill out this form. Fill out your group name. Below, there are two slots for specifying the group members. The first slot will automatically be filled with the NetID of the user who is logged in. In the second slot, write the NetID of the other group member. Even if you want to work alone, you need to create a group for yourself.

Attention: If the other person has never logged in to InfoBase with her NetID before, write instead her bk-account name (student number including b or bk, for example, b1234567)

Choose an existing building to analyze from the electronic plannanmap

You should analyse the same building as the one you are analysing in your design project.

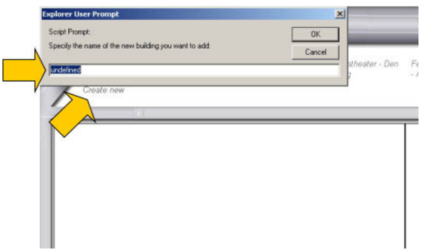
After you create a group, you will get to the "Workshop1 - analyse bestaand ontwerp" page. Click on the first link to open the electronic plannenmap.



The electronic plannenmap will open in the same window. You will use this interface to choose a building to analyze, or if you want to analyze a building that is not included in the plannenmap for museums, you can create a new building in the electronic plannenmap and upload images for this new building. The new buildings will be visible for usable by all students. If your building is not included in the book, come to the workshop with the digital images of your building.

Select a building from the plannenmap.

If you are analyzing a building that is not included in the plannenmap, click on "create new" and create a new building in the electronic plannenmap.



In order to upload images into any building, scroll to its name in the upper frame and click on the link. At the lower frame, click on "upload" and fill out the form that appears.

Browse through the pictures by dragging and dropping them into the two frames above the thumbnails.

Pick 2 planmiddelen that are important for the building you have chosen

You have already started your analysis in the design project. You already have an idea about what are the most important planmiddelen that led to the design of your selected building. The goal of this exercise is to express digitally two analyses of the selected building analyzing the building according to the two planmiddelen.

Some examples of planmiddelen are listed under the relational metadata: "constructive opbouw, flexibiliteit in gebruik, functionele opbouw, licht als planmiddel, maatsystematiek, relatie met bestaande bebouwing, relatie met omgeving, routing, ruimtelijke opbouw". You can analyze these planmiddelen using instruments and structural characteristics of these planmiddelen such as the keywords listed under constructive metadata: "afwerking, draagstructuur, installatie, kleur, massa, materiaal, omsluiting - dak, omsluiting - gevels, opstelling - centraal, opstelling - collage, opstelling - grid, opstelling - lineair, proportie, ritme, ruimte - administratie, ruimte - circulatie, ruimte - entree, ruimte - horeca, ruimte - service, ruimte - tentoonstelling, schaal, symmetrie, textuur, transparantie, uitvoering".

These 2 analyses need to be submitted using the DAN Toolkit.

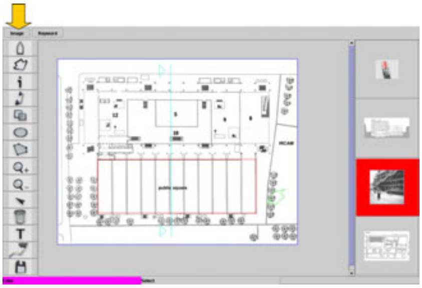
You do not always have to use plans for analyses. Sometimes using a site plan, a secion, an aerial photo or another photo is a lot more appropriate. We expect you to be creative and expressive in your analyses.

Analyze the building according to these using Design Analysis Network (DAN)

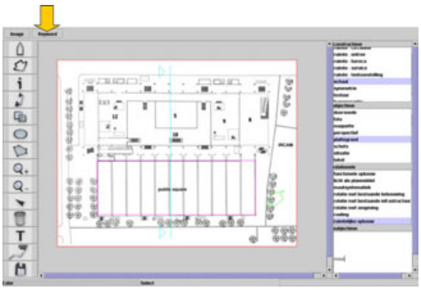
The picture that is in the left picture frame will be the one that gets loaded into the DAN Toolkit. Click on "Menu" and then "Toolkit" to open the DAN Toolkit.

Below are some images of the DAN Toolkit. Click here to go to the manual for this toolkit.

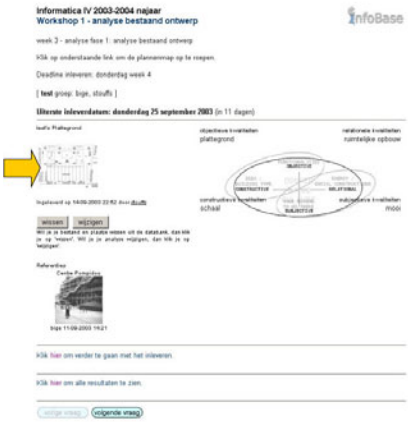
Place markers on the plan and relate these markers to other pictures from the same building. The tool will not save the analysis unless another image is related to each marker. The marker that is missing a picture link will be highlighted.

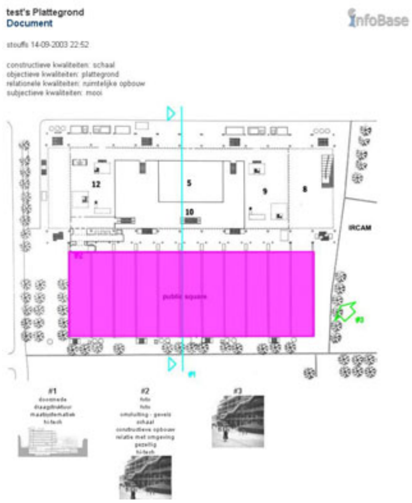


Attach metadata to all markers on the plan and also to the main picture that you are analyzing. At least one keyword from objective, constructional and relational groups must be attached to each marker and the entire picture. Also type a subjective keyword for all of these. Subjective keywords can be freely defined. The tool will not save the analysis unless metadata is attached to all markers and the entire picture. The marker that is missing a keyword will be highlighted.



When you click on the "save" button in the toolkit, you have submitted your analysis. The resulting page will be as follows.

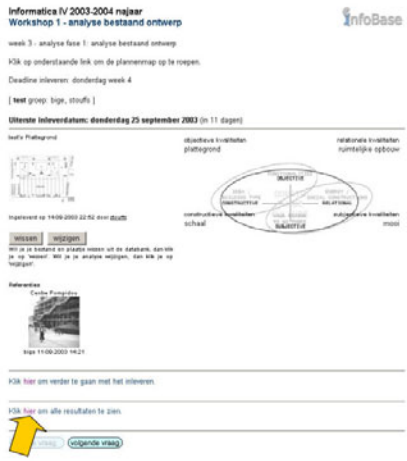




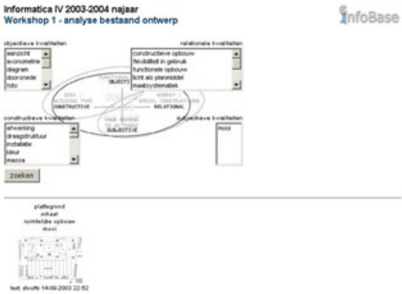
When you click on the thumbnail of your analysis, you will get the rendered version. You can delete your analysis by clicking on the "wissen' button. If there are things you want to change, you can open your analysis in the toolkit by clicking on the "wijzigen" button and modify it. You can click on the link that reads "click here om verder te gaan met het inleveren" to do your second analysis in the same way as the first one.

Practice how to search the database using the metadata

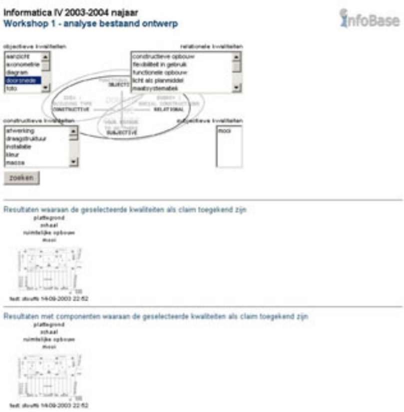
Follow the link to look at all the results in the database.



Here, you can see an overview of all the metadata grouped under the 4 dimensions: Constructive, Objective, Subjective, and Relational. You can select one or more entries from each group and click on the "zoeken" button to search for the submissions which contain the selected metadata attached to the main picture or any of the markers. The search will be an AND search between the dimensions, meaning the search result will contain only the analyses that contain all of the selected entries. If you have more than 1 entry selected within a single dimension, the search engine will carry out an AND search, meaning it will select the analyses that contain any of the selected entries. If no entry is selected, all results will be shown.



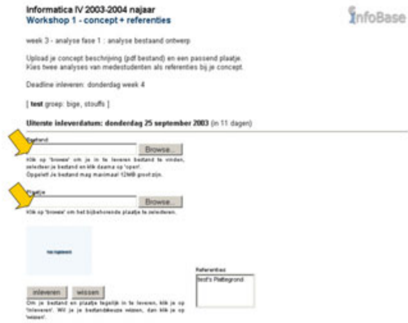
The search results are grouped under two sections: according to the main image (resultaten waaran de geselecteerde kwaliteiten als claim toegekend zijn) and according to the markers in the image (resultaten met componenten waaran de geselecteerde kwaliteiten als claim toegekend zijn).



Part 2:

Write a short text (250 words) describing your design idea for your own design

You have already done a volume study of your design for a museum. Therefore, you already have a broad idea about how you want to do your design, in abroad sense. For example, will your design have a central plan? Will it be a juxtaposition of two massive volumes? Or will transparency play an important role? You will write a short text (250 words) of your design concept. You will submit this text as a PDF file by going to the submission page of question 2. You will also supply a preview thumbnail image for this PDF file.

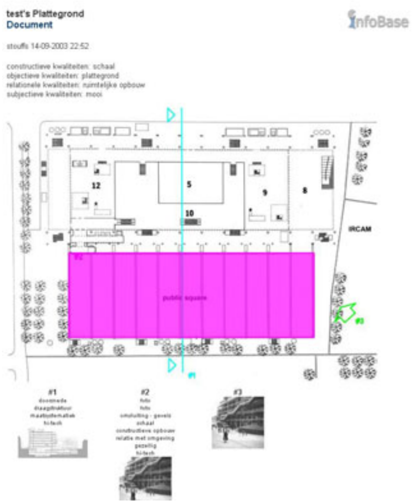


Choose two analyses as references to your design concept

Search the database by using the metadata. Find two other analyses that somehow strengthen the oint that you make the point in your design concept text. For example, the reference may have a same or similar planmiddel that you want to use in your design. Or, the reference may be contradictory to your design concept, etc. **Once you choose the references using the search interface, pick them from the list provided in the <u>submission page of question 2.</u>**

Print your analyses, your design concept text, and the two references and show these to your design teacher during critiques

Open your analyses and the reference analyses by clicking on the thumbnail. Print the analysis using the print functionality of your browser.



How to work from home

The InfoBase system and the analysis tool are web-based, and can be reached from any location from computers with an internet connection. Below are some points of consideration.

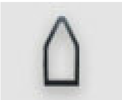
- If you are using Windows XP as operating system, you will see only two fields at the login screen, username and password. Fill in your username as **bk-stdnt\student number**.
- The analysis tool is a Java applet. If you cannot open the tool, it may be that java virtual machine is not installed on your computer. You can download it from http://java sun.com/getjava/index. html.
- If you still cannot open the analysis tool, it may be that the settings of your internet browser (Internet Explorer) are not corrent. Open Tools -> Internet Options from the menu. Click on the Advanced tab. Scroll down. There is a group of settings with the title Microsoft VM. Check all three options under this group. Quit and restart your browser.

Deadline of submission: Thursday, Week 4.

Description of the DAN Toolkit:

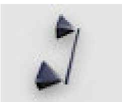
The two buttons "image' and "keyword" at the top denote the two model of the toolkit. It is possible to link images to markers in the image mode, and to attach keywords to markers and the main image in the keyword mode.

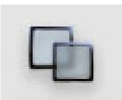
Select the image below to:


denote a linked picture at that location and direction


denote a linked elevation at that location and direction


denote an annotation about the whole analysis or a part of it, e.g., the color classification that is used, a piece of information about the analysis done, or a piece of information about a part of the building

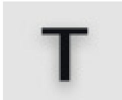

denote a linked section at that location and direction


draw a filled rectangle


draw a filled ellipse


draw a filled polygon, right mouse button click closes the polygon (do not make more than 11 vertices!)

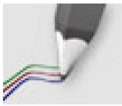

select an already drawn shape


add a short text to a selected shape (max. 100 characters): select shape, click on this icon, add your text (pasting is possible)

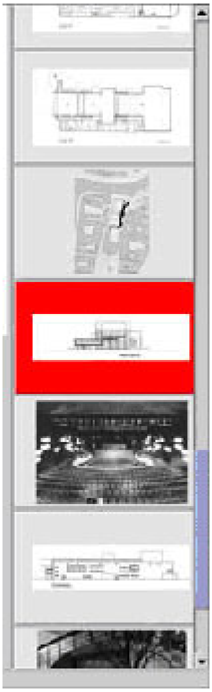

zoom in the whole image


zoom out


delete the selected shape


repeatedly to select one of the predefined colors


save the composition and quit application



Click on one or more images from the right hand side of the application in image mode (the background will become red) to link them to the selected shape

Select keywords from the right hand side of the application in keyword mode to link them to the selected shape or the whole image. The keywords in the objective, constructive and relational dimensions are fixed. You can freely define one subjective keyword per marker.

B

SCHEMATIC OVERVIEW OF THE INFOBASE DATABASE

This appendix contains the schematic overview of the InfoBase database. The prototype applications BLIP, DAN, and DesignMap described in Chapter 5 have been implemented using this database and the scripts developed for it.

InfoBase system has a single database that supplies a simple, open structure for managing documents and metadata. The database model mainly distinguishes 8 different object classes, these are *projects*, *documents*, *links*, *persons*, *groups*, *access rights*, (metadata) *dimensions* and *marks* (Figre B.1): projects define workspaces, e.g., courses; all content is defined as documents, documents can be linked; persons can be grouped and access rights defined per group, access rights apply to the entire project or to a specific document; (metadata) keywords are distinguished (and possibly structured) by dimension; marks define assessment marks or grades. Both documents and links are distinguished by type. Document and link types serve a categorization that is mainly semantic but may be used by the interface modules to distinguish functionality. Figure B 2 presents a detailed view of the database model.

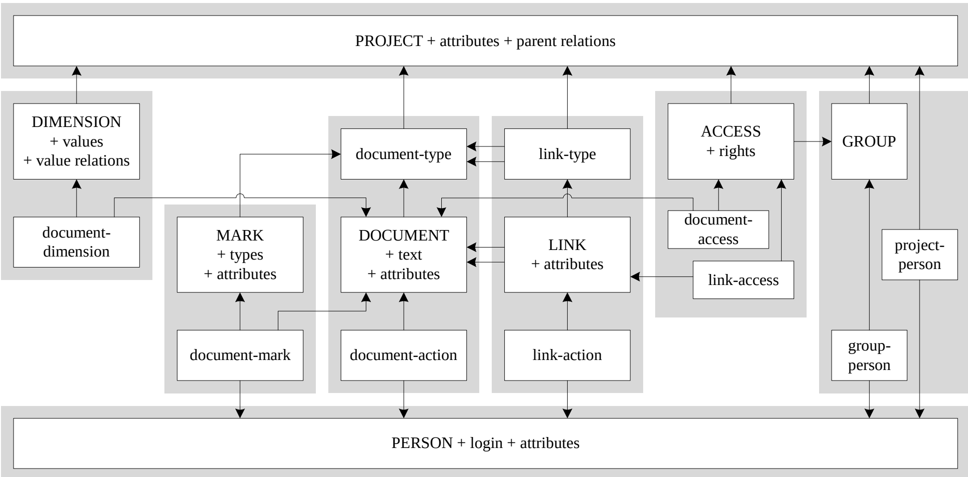


Figure B.1: The abstract InfoBase database model. The shaded areas distinguish the eight main object classes (the respective class names are in all capital letters). For clarity, some tables are grouped together and presented by a single box (each plus sign refers to a separate table in the actual database model).

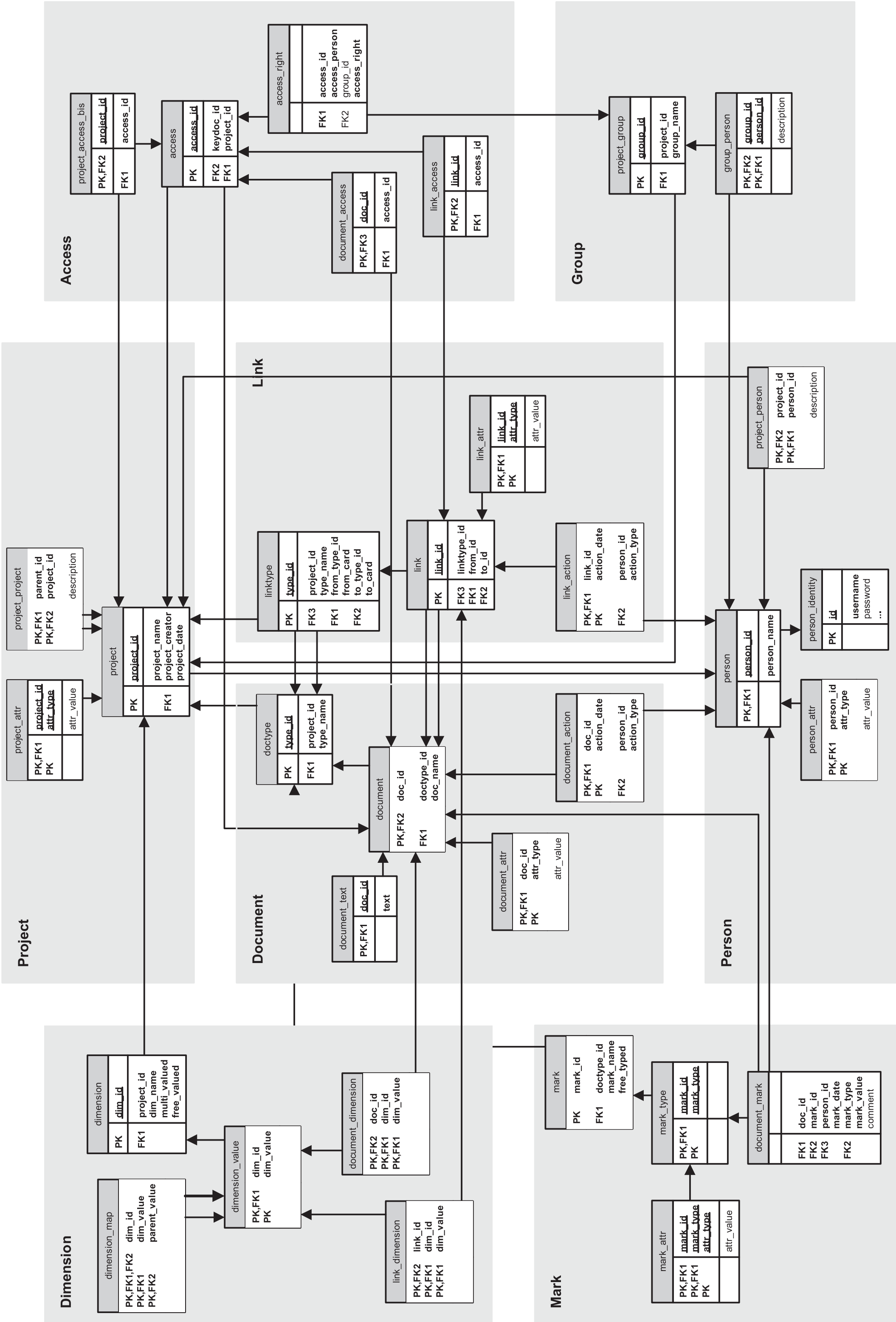


Figure B.2: The detailed InfoBase database model.

C

EVALUATION REPORT OF DESIGN ANALYSIS NETWORK

This appendix contains the evaluation report of Design Analysis Network (DAN), a prototype application developed in order to create a digital architectural analysis library (see section 5.2). This application was used in the second year design studio of the education in the B.Sc. program at the Faculty of Architecture, Delft University of Technology. This report was prepared in December 2003 by Jelle Attema (Laboratory of Work and Interaction Technology, Faculty of Technology, Policy and Management) , Evren Akar (Faculty of Technology, Policy and Management), and Bige Tunçer (Faculty of Architecture), all from the Delft University of Technology. This report has been included in this appendix in its original form.

EVALUATION REPORT OF DESIGN ANALYSIS NETWORK

December 2003

About the report

This report consists of:

- Introduction, research questions and evaluation setup:
- Results
- Appendices
  - Quality of the results: the number of new findings per session and the weight of the results: the number of remarks related to a topic.
  - Detailed findings

The results part shows the categories of results and suggestions for solutions. If the background of the conclusions is not clear, the list of detailed findings may help to check interpretation.

Introduction, research questions and evaluation setup

Design Analysis Network is an internet-application developed by the Chair Design Informatics of the faculty Architecture of the Delft University of Technology. The application is used for presenting the results of architectural analysis of a building. The results of the analysis are stored in a database and metadata is automatically added.

The metadata make it possible to search on characteristics of buildings and details. In that way the database can also be used during design: buildings that are related to a design on a more abstract and conceptual level can be retrieved from the database and can be used for inspiration.

The application is used second half of 2003 by architecture students for an analysis assignment. From the feedback of the students it appeared that the possibilities of DAN and the use of DAN were not clear. A possible reason was the embedding of DAN in the educational process (lack of fit). Usability issues apparently played a role.

The goal of the evaluation was to find out the nature of the problems students had with DAN, with its use and its usefulness, and how it fits into the educational process.

Setup of the evaluation

Five sessions were organized with students that followed the course and had tried to do an analysis with DAN in the WIT-Lab. A student and a project member of DAN were seated in a separate room. In the observation room the activities of these two people could be followed and it could be seen how they used the computer. Two researchers of the WIT-Lab were present to take notes and keep an event log. Also in the observation room, another project member and a number of teachers of the analysis course were looking at the session. They were invited to comment on their observations. Their remarks were also put in the event log.

The first part of a session the student explained and demonstrated to one of the project members of the DAN-project (involved in the design and development of DAN but also involved in the educational process) how they had used DAN. They redid one of the assignments they did when doing the course.

The second part the session project member explained the student how DAN was intended to be used. Then the student did a second assignment trying to use DAN as proposed by the project member. This assignment was also related to an assignment they did before. To illustrate the added value of the metadata the student did a search assignment (related to his/her own design).

The sessions (the event log and the video of the computer screen) were discussed with the DAN project members. They were invited to formulate “findings”: remarks on the usefulness and usability of DAN, either positive or negative.

The findings were categorized and summarized by the researchers of the WIT-Lab. The findings belonging to the same category were summarized and based on these summary problems and solutions are suggested. The different chapters in this report are formed by these summarized findings.

Summary of the conclusions

It appears that DAN is valued very much by students once they understand what it can do. And that is also is the heart of the problem: when using DAN without a (small, proper) introduction, students and teachers do not understand where DAN is about. A number of issues contribute to this problem:

1. When just looking at DAN it does not become clear to students and teachers what DAN is and what it can contribute to analysis. Different teachers have different approaches to analysis: it is not clear how different ways of analyzing fit into DAN. The choice and definition of keywords is not clear to students. Good examples/cases (also showing a way of working) may help.
2. Once students have seen how searching the DAN-databases can help in the conceptual stage of design (get new ideas, inspiration) they get enthusiastic. Only a very small introduction seems enough to let them understand what DAN can contribute. Such an introduction was given in the design studio but apparently was not sufficient. This instruction could be part of DAN.
3. The role and value of metadata is not understood by students: the function of the keywords and the meaning of the categories are not clear. This results into sub optimal and improper use of keywords. Several suggestions are done to improve the keywords and the naming of the categories.
4. There are several fundamental usability issues when using DAN: even though all students used DAN before several (serious) problems occurred resulting into inefficiency and errors. Part of these usability issues may easily be solved.

An analysis of the new findings per sessions (see section “The Quality of the evaluation”) shows that after the first two sessions the number of new findings (new information) per session decreases very strongly. The conclusion can be drawn that adding new sessions probably would not add much new information.

Results: summary, problem areas

Problem Area 1: Integration of DAN into educational program

It appeared during the evaluation that students did not have a clear idea about the role that DAN could play during analysis. When students started to work with DAN they had to find out themselves about the goal of DAN and how DAN fits into the design course.

Summary:

- Expectations of teachers on analysis are often not clear to the students: DAN supports several approaches to analysis and is primarily a means of communicating the results of analysis. However: when it is not clear to a student what the teacher expects of analysis, the use of DAN will not be clear either. DAN should therefore illustrate how DAN can be used for analysis and concept development.
- Many teachers do not have an idea how DAN fits into their design-course: DAN should explain to teachers how it contributes to analysis and can help in concept development. However: DAN should also explain students when teachers do not succeed in communicating the relation between DAN and design-courses or when they miss a part of the explanation.
- When teachers do not have a good idea on the value of DAN this may easily lead to double work for the students: first prepare their work in DAN, then do it again in Photoshop or some other program.
- For full integration of DAN into the curriculum it is important that DAN refers to the semester book and vice versa.
- The added value of using keywords (metadata) is not clear to teachers and students: the use of keywords is not part of the course.
- The keywords as they are defined in the present version of DAN are not clear to teachers: why does a keyword belong to a certain category, some keywords seem to overlap. It is important that the keywords used in DAN are created by the teachers during the course. The meaning of the metadata should be shared by the teachers.

Solutions:

Expectations of teachers not explicit:

- DAN can instruct students that there are several approaches to analysis (some examples) and that it is important to know what approach is favored by the teachers. It should also give an indication how DAN can support several approaches.
- Also teachers should have a separate entrance in which the importance of communicating expectations is stressed and it is indicated how DAN supports several approaches to analysis.

Relation between design course and DAN:

- DAN should explain to students how it fits their course.

Prevent double work:

- DAN should give suggestions how it can be used to make print-outs, how it can be used to prepare presentations.
- Teachers should be encouraged to use DAN when students present work
- Teachers should give example how to hand in work using DAN. When they do not want to use DAN should show students how they can use screen dumps, Photoshop to make quickly and easily a printout of their work.

Semester book:
- DAN should be part of semester book and refer to semester book.

Keywords:
- There should be a separate session together with teachers in which the use and value of keywords is explained and in which teachers can come up with a categorization that is shared by all teachers.

Problem Area 2: The added value of DAN

The evaluation sessions had two parts: one part in which the student explained to the teacher how DAN was used. The second part the teacher told the student about the intentions of DAN and then the student tried to use DAN in the intended way. It appeared that DAN itself does not communicate clearly to students and teachers what the added value is.

Summary:

The evaluation showed that DAN has four valuable aspects:

- Being exposed more intensively to the work of others (architects, students, teachers) contributes to the quality of the educational process. DAN may contribute significantly to this goal.
- DAN teaches students a vocabulary to communicate about buildings (analyze buildings and communicate the results of analysis).
- DAN provides a means of communication itself: it helps students to express their opinion/story about a building after doing an analysis.
- DAN helps students to develop a design concept, to enrich the design concept and finally to get ideas how the design concept can be realized.

One of the problems of the present setup of DAN is that its added value only appeared after its goals were explained to students and students were guided to use it.

- The user does not receive **guidance** when trying to grasp the added value of DAN. The user is not shown how to proceed when trying to use DAN for analysis. The role of meta-information is not clear. The value of the database is not clear.
- The **contents** of DAN do not demonstrate particularly well the added value of DAN. The reason is that all work, whether good or bad is shown in DAN. This may have two side-effects. One is that students will not understand and grasp the added value of DAN. Second is that because of the quality of the content students will not be

motivated to hand in good work themselves. They do not see the added value of doing a good job.
- However: once a student understands how DAN is used, they are able to work with it in the intended way.

Solutions:

Communicating the added value of DAN:

- Each year the best contributions in the DAN-database should be filtered: in an assignment students should be demonstrated how they can make use of these contributions when thinking about design concepts and doing analysis.
- There should also be a small number of case-studies that illustrate how DAN can be used for concept development (enriching design concepts) and during analysis of buildings.
- DAN should provide a guide that demonstrates how it can be used for concept development, for analysis etc. It is important that this guide is not completely textual but also uses the means of DAN to communicate the added value.
- Specific assignments (together with an instruction how to proceed) may help students to understand the value of DAN. For example: a search assignment to elaborate their own design concept or an assignment to describe (after handing in their design concept) how their own design concept relates to other buildings.

Problem Area 3: Procedure for doing analysis in DAN

In the previous paragraphs it was discussed that the intentions of DAN and how it may fit into the design course is not clear. However: once a student grasps the intentions, there are still some problems on the way of working: how should DAN be used?

Summary of findings: How should DAN be used?

Although during evaluation it appeared that with a little guidance students grasp the goals of DAN easily, students had difficulties in finding out these goals themselves. The interface does not provide any guidance or examples how to start and how to go on.

In detail:

- It is not clear to students how DAN plays a role in analysis (preparing analysis, doing analysis, communicating analysis).
- It is also not clear that DAN proposed a specific method for analyzing architecture: the goals of the architect, the means and their criticism.
- It is not clear how DAN can be used for several different ways of analysis: the viewpoint of the architect, the viewpoint of the student or the viewpoint of the users.
- DAN does not provide a way of working and that students have difficulty in finding a way of working themselves. DAN should provide more guidance on specific aspects of analysis.
- Finally it is not clear how DAN can be used to present and communicate the results of analysis.

Solutions:

- There should be several case studies, each showing a specific aspect of DAN.
- Some clear case studies that demonstrate how students can proceed when doing their assignment.
- Content of the database of such quality that students can grasp the added value of DAN better.
- The difficulties with specific aspects DAN should be addressed explicitly in the case studies.
- Design teachers should explain to students how to do analysis.

Problem Area 3: How to support students

Summary of findings and solutions

1. Students need guidance on how to proceed when entering DAN. Creating a group, the assignment, how to proceed after a picture is selected, when to choose keywords, when to choose links. Every step should be made explicit.
2. Students need some help how to detect the goals of an architect: for example: start with means of the architect and then ask: why did he use these means. From the goal back to the means.
3. Students need guidance on how to use the means DAN provides (keywords, links, markers) to communicate the results of analysis.

    - The purpose of the different tools is not clear: when do you use an arrow
    - Role of information boxes in explaining ideas is not clear to student
    - Not clear when keywords should be used

4. Students need guidance on the role of programs like Photoshop, and that you upload images that illustrate your point. Students thought that DAN was mainly to prepare analysis or for drawing.

Problem Area 4: General Usability Issues:

Summary of findings and solutions

During evaluation it appeared that two usability issues seriously contribute to difficulties in using DAN:

1. Students are afraid that program crashes: first adds all pictures, then all markers and then all links.
2. Students are forced to save a correct and complete analysis. However: DAN does not provide an overview what is done and what still has to be done before work can be saved when working on the analysis. The feedback on errors is not very clear: sometimes the elements on the screen (causing a problem) are so small that the student overlooks the element and cannot detect the cause of a problem. Students cannot save work in between and take up your work next

day. It means that now and then students just shut down their computer losing all their work.

Problem Area 4: Usability issues: use of keywords

Summary of findings and solutions

The **meaning of the keyword groups** is not clear:
- Relational: main goal of architect;
- Constructive: means to reach this goal.

Also when the meaning is explained it appears difficult to remember the meaning.
Suggestion: more specific naming of keyword categories.

The order of the **keyword categories** is also not logical:
- first the goal of the architect should be analyzed (relational),
- then the constructive keyword should be selected (selection of means to reach the goal).

Objective keywords: objective keywords are a property of the file that is uploaded. Suggestion: force the selection of an objective keyword when uploading a file.

Subjective keywords:
- It is not possible to make use of a previously defined subjective keyword: each subjective keyword is therefore forced to be unique. Each keyword just leads to one unique record in the database. It should be possible to select a previously defined keyword.
- It is not clear when and how a subjective keyword should be used (and when a comment for example). An example can clarify the choice of a keyword.
- Subjective keywords should be moderated regularly: now some strange keywords are in the list.

Other usability issues related to keywords::
- The use of shift and control for multiple selection of keywords will not be clear for all students. This should be made explicit.
- You cannot see which keywords are selected: confusing when a keyword is selected in the list while it is not visible anymore.
- It would be nice, while selecting keywords, to see the picture for which keywords are selected.

Problem Area 4: Usability Issues: use of the markers

- If the program gives feedback on missing links, the marker is selected. First it has to be deselected before the student can proceed. Leads to many errors.
- When selecting a marker, the marker remains selected. First the selection-tool has to be selected to deselect it. As a consequence users often put markers by accident. After placing a marker the selection tool should be activated automatically
- Difficult to place an arrow/marker directly on the right place.

- You cannot change the place of an arrow once it is put on the drawing.
- It should be possible to add information or a marker without adding a picture to it.
- Without adding texts to links it will not always be clear to other readers why the link is there.
- It is not obligatory (possible) to add information to a link between a marker and a picture.
- Student wants to link a picture to a group of markers. It is not possible.
- When two markers are on top of each other, you cannot access the lower marker anymore.
- It is not clear how much text the information marker can store. It is also annoying that you cannot add new lines and indents to annotation texts.
- You cannot change color of markers afterwards.
- When a form is used as a marker (rectangle) it is shown in toolkit as a massive block. Blocks view. In printview it is semi-transparent. It is confusing.
- Difference between view and elevation is not clear (tools).
- Texts are not always readable: default text should have a contrasting color.

Problem Area 4: Usability Issues: printview

Findings

Relation between keywords on top of printview and the rest of page is not clear.

Thumbnails in printview are too small. When you enlarge the thumbnail you do not see anymore how the picture is linked to the complete overview

In printview **texts** are not shown.

Pictures are shown in printview in **order** in which they were added. When explaining for example routing it should be possible to influence the order in which pictures are presented.

The **numbers are missing** in printview

You cannot read associated text when looking at a picture associated with a marker (in printview). Text should have **contrasting color**.

Problem Area 4: Usability Issues: search

- There is no feedback on which keywords is searched and how they are combined (and, or)
- It should be clear which keywords are selected (in a list)
- Possibility to search on a specific architect or name of a building

Problem Area 4: Usability Issues: other

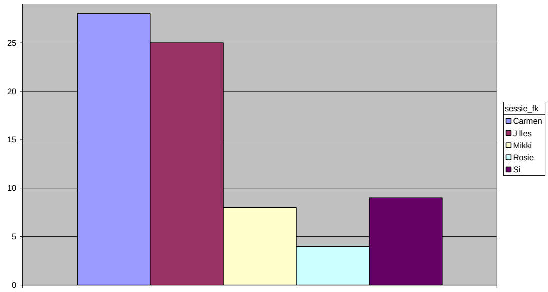
- The document has no title: you cannot see any information about the building that is analysed
- You cannot see which building is analysed in DAN (not in plannenmap, not in printview)
- Pictures should be scaled in such a way that they fit into their frame.
- students are often graphically oriented: a map is not a good visualizer. When selecting a map as a startingpoint for analysis, it should be possible to add a thumbnail that illustrates what the analysis is all about
- when selecting a plan or sketch as a basis for analysis, a picture should be added that illustrates the essence of the analysis.
- a scroll bar is necessary next to keywords: now only in full screen mode subjective keywords can be added
- The cookie expires after some while. Is confusing to student: what happens, do I lose work, what should I do. Should be better feedback.

Problem Area 4: Usability Issues: bugs

When you add a subjective keyword and you don't save it explicitly (press enter) it gets lost.

Appendix 1: The Quality of the evaluation

Number of new findings per session

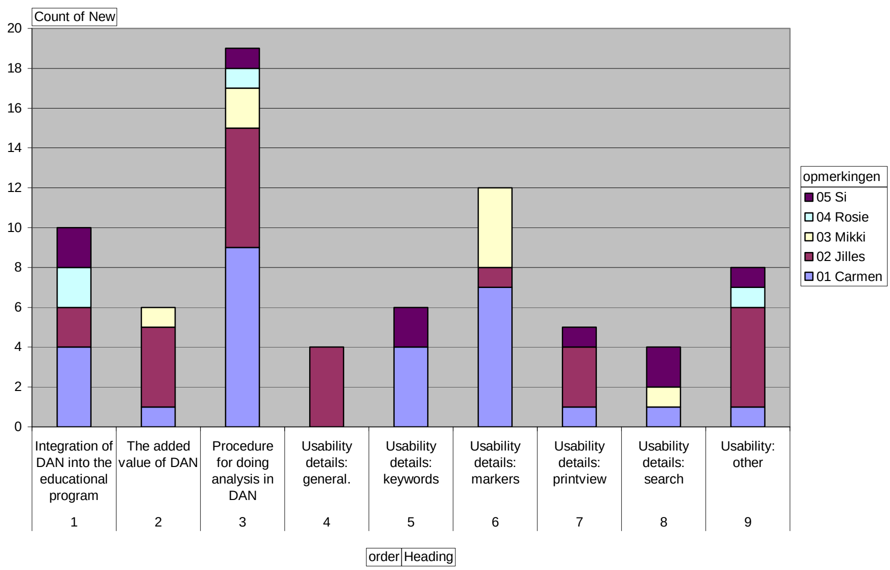


Graphic 1: showing the number of new findings per test session.

The table shows that the number of new findings per test session reduces strongly after two sessions.

When after several sessions still many new findings are found one can argue that new sessions would probably add new information to the evaluation. However: in this case the number of new findings drops significantly after two sessions: adding new session probably would not have added any new insights.

Weight of the findings



Graphic 2: shows for each of the 9 evaluation themes the number of new remarks each session.

The graph shows that the issue “how to do analysis” in DAN receives the most new remarks (19). Apparently it is a very central issue for the evaluators and the topic apparently has a lot of aspects: new aspects are added even in the last three sessions..

The usability issues related to the markers receive a lot of attention (37 from 72 new remarks): it is apparently a very central usability issue.

There are roughly as many usability issues (37) as issues on how to use DAN (35).

Appendix 2: detailed report of findings

Integration of DAN into the educational program

Name	Total Of Name	1	2	3	4	5
expectations of teacher	10	3	3		2	2
meaning of keywords	1		1			
time/double work	1	1				

opmerkingen	Total Of Name	01 Intro	02 Stud. analysis	03 Guided analysis	04 Instruction	05 Search in database	06 Evaluation
01 Carmen	4			4			
02 Jilles	4		2				2
04 Rosie	2						2
05 Si	2	2					

In all sessions the issue of the integration of DAN into the educational program was discussed: during the introduction (where it was a specific question), during the students analysis and also when the student did guided analysis. Often students remarked after explanation of DAN that they did not know that this was the purpose of DAN.

Findings

Integration education program: expectations of teacher	
Expectations: students do not know what the expectations of the teacher are when doing analysis in DAN	01 Carmen
Teachers have no clear how DAN fits into their design-course. DAN should be self explanatory in explaining usefulness and how it can be used	01 Carmen
Student would like to make two analyses: one from the viewpoint of users, one from personal viewpoint. Is possible within assignment. Maybe part of example	01 Carmen
Student handed in analysis: worked hard on it. Teacher was not satisfied. Student did not know why: expectations not clear	02 Jilles
Teachers should be encouraged to use DAN when students present their work. DAN should suggest and give example how to use screen dumps and photoshop to present analysis when DAN environment is not used and a teacher wants a printout	02 Jilles
Concept of keywords: not properly introduced during course. Is new to students when they enter DAN. Should be explained	02 Jilles
Should be separate explanation of DAN for teachers and for students.	04 Rosie
More people are required to work on DAN and education.	04 Rosie

The description of DAN should also refer to the semester book	05 Sigrun
Is not clear what the teacher gets: is it DAN or a printout. Should be explained to students how they can present results in DAN to teacher.	05 Sigrun
Integration education program: meaning of keywords	
The keywords used in DAN: their meaning is not always clear, also not always clear why they are in a certain category (means, goals). Explanation is required and also teachers should agree on the way keywords are interpreted	02 Jilles
Integration education program: time/double work	
Thinks that during design classes there is not enough time to learn to use DAN.	01 Carmen

The added value of DAN

Name	Total Of Name	1	2	3	4	5
usefulness	10	1	5	1	1	2

opmerkingen	Total Of Name	01 Intro	02 Stud. analysis	03 Guided analysis	04 Instruction	05 Search in database	06 Evaluation
01 Carmen	1			1			
02 Jilles	5			1			4
03 Mikki	1	1					
04 Rosie	1						1
05 Si	2	1		1			

Remarks on the added value of DAN are found in all sessions, but especially in sessions 2 and 5. Remarks are related to the search assignment.

There are no specific conclusions that can be based on this pattern.

Findings

Procedure: added value of DAN, usefulness	
DAN should make explicit how it can aid to develop a design concept, to enrich it and to get ideas how a design concept can be realized.	01 Carmen
The contributions of students should be filtered after each semester so that only good work is in the database	02 Jilles
Value of DAN should be made clear: helps you to bring your point across in an easy and quick way	02 Jilles
When students know how their work relates to the other work in the database and when they have experienced the added value of DAN they will feel	02 Jilles

responsible to hand-in good

DAN should give feedback how the selection of certain keywords makes that the design concept of a student relates it to other buildings: for example. For example: your design concept is related to the work of x and building y.	02 Jilles
A search-assignment based on design-concept can illustrate the value of DAN for elaborating a design concept	02 Jilles
Student expects that DAN will expose students more intensively to the work of others	03 Mikki
To illustrate the value of search students should be able to find their own design concept using keywords describing their concept.	04 Rosie
The added value of doing analysis in DAN is not very clear	05 Sigrun
Essence of learning vocabulary when doing anaysis is listening to others. DAN should assist in listening to others	05 Sigrun

Procedure for doing analysis in DAN

Name	Total Of Name	1	2	3	4	5
procedure: how to detect goals of architect	1		1			
procedure: how to proceed after picture is selected	1			1		
procedure: how to start	4	2		1		1
procedure: limited time	1				1	
procedure: role of keywords	2		2			
procedure: role of keywords, links, markers in communicating results of analysis	1	1				
procedure: use and function of different tools is not clear	2	1	1			

opmerkingen	Total Of Name	01 Intro	02 Stud. analysis	03 Guided analysis	04 Instruction	05 Search in database	06 Evaluation
01 Carmen	9		3	6			
02 Jilles	9		3	4			2
03 Mikki	2		2				
04 Rosie	1						1
05 Si	4	2	1	1			

The issue how to do analysis in DAN was a central issue during the evaluation: students do not understand how to start in DAN, how to use the tools..

Findings

Procedure: how to detect goals of architect

Students need some help how to detect the goals of an architect: for example: start with means of the architect and then ask: why did he use these means. From the goal back to the means.	02 Jilles

Procedure: how to proceed after picture is selected	
Not clear how to proceed after a picture is selected: when to choose keywords, when links. Should be made explicit not clear how to proceed after a picture is selected: when to choose keywords, when links. Should be made explicit.	03 Mikki
Procedure: how to start	
Guidance: needs guidance how to proceed when entering DAN. First create a group, description of assignment, should be more clear.	01 Carmen
Guidance: role of upload is not clear. When do you upload a picture.	01 Carmen
It should be clear what the starting point is in a presentation of an analysis in DAN: a map, a picture, a sketch made in Photoshop etc.	03 Mikki
When describing the procedure for analysis: it should be clear that students should collect missing pictures	05 Sigrun
Procedure: role of keywords	
Not clear when keywords should be used.	02 Jilles
Explanation of role of objective keywords: contributes to search process	02 Jilles
Procedure: role of keywords, links, markers in communicating results of analysis	
The role of the different tags, the keywords is not clear: when do you add texts, when do you use a subjective keyword.	01 Carmen
Procedure: use and function of different tools is not clear	
The purpose of the different tools is not clear: when do you use an arrow,	01 Carmen
Role of information boxes in explaining ideas is not clear to student.	02 Jilles
Procedure: when and how DAN is used during analysis	
Definition of analysis: analyze how a building functions. From different viewpoints: for example environment, those who live there etc.	01 Carmen
Main function of DAN: presentation of the results of analysis. In such a way that also others can use the results to enrich their design-concept	01 Carmen
is not clear to the student how DAN can be used to illustrate their ideas about the goals of the architect, the means and their criticism	01 Carmen
it is not clear how the keywords are used in relation to the markers, links to tell the story the student wants to tell about the building	01 Carmen

After explanation of the purpose and method of analysis the student states that this was not clear when she started to use DAN	01 Carmen
Student interprets analysis as a kind of documentation and illustration what the building is about.	02 Jilles
The role of programs like Photoshop is not clear, and that you upload images that illustrate your point, is not clear to student. Example should illustrate this	02 Jilles
role of Photoshop and the role of keywords, links and adding information should be made explicit in a number o examples	02 Jilles
In example it should be made clear that using DAN is to a certain extent the end of analysis: when you know what you want to tell. At the other hand DAN provides a method for analysis and helps to reach these goals.	02 Jilles
Student thought that DAN was mainly to prepare analysis.	02 Jilles
Student wants to tell a story about the building. It is not clear how the means (keywords, links, markers) are used to tell this story. Keywords do not tell what she wants to tell.	05 Sigrun
It is not clear how you can tell your story using DAN	05 Sigrun
Students easily think that DAN is for drawing.	05 Sigrun

Usability details: general

Name	Total Of Name	1	2	3	4	5
usability - crashes - testing	1		1			
usability - feedback	1		1			
usability - markers - missing links	1		1			
usability - saving	1		1			

opmerkingen	Total Of Name	01 Intro	02 Stud. analysis	03 Guided analysis	04 Instruction	05 Search in database	06 Evaluation
02 Jilles	4		4				

One tester had experienced a lot of difficulties with DAN and reported them very precise. He identified a number of usability issues, not reported by the others.

Findings

Usability – crashes – testing	
Student is afraid that program crashes: first adds all pictures, then all markers and then all links.	02 Jilles
Usability – feedback	

No overview what is done and what still has to be done before work can be saved	02 Jilles
Usability – markers – missing links	
If the program gives feedback on missing links, the marker is selected. First it has to be deselected before the student can proceed.	02 Jilles
Usability - saving	
You cannot save work in between and take up your work next day.	02 Jilles

Usability details: keywords

Name	Total Of Name	1	2	3	4	5
usability - keywords - use of shift and control	1	1				
usability - keywords - meaning of categoires	6	2		4		
usability - keywords - objective keywords	1					1
usability - keywords - order of keywords	3	1	1	1		
usability - keywords - select keywords	2	1				1

opmerkingen	Total Of Name	01 Intro	02 Stud. analysis	03 Guided analysis	04 Instruction	05 Search in database	06 Evaluation
01 Carmen	5			5			
02 Jilles	1			1			
03 Mikki	5			5			
05 Si	2		2				

The role and use of the keywords is an important issue during guided analysis: then the teacher explains the student how keywords are supposed to be used. Most of the remarks on the use of keywords are during guided analysis. Maybe this indicates that the use of keywords is less a problem for students until they understand how to use the keywords.

Findings

Usability – keywords – use of shift and control	
It is not clear to the student that keywords can be combined within a category.	01 Carmen
Usability – keywords – meaning of categories	
The meaning of the keyword groups is not clear: relational = main goal of architect; constructive: which means to reach the goal	01 Carmen
Meaning of keyword categories is not clear (relational, constructive)	03 Mikki

Meaning of keywords is not clear	03 Mikki
Usability – keywords – objective keywords	
Why not couple objective keywords when drawing is uploaded: is fixed.	05 Sigrun
Usability – keywords – order of keywords	
The order of the keyword categories is not logical. First goal, then means, then subjective. Objective is separate	01 Carmen
Order of keyword categories should be changed: first goal, then means, then subjective.	02 Jilles
Wants a different order of keywords: indicates how to go on.	03 Mikki
Usability – keywords – select keywords	
Use of shift and control when selecting keywords is not clear to all students.	01 Carmen
Would be nice that you can see picture while selecting the keywords belonging to it	05 Sigrun

Usability details: markers

Name	Total Of Name	1	2	3	4	5
usability - color of texts	1	1				
usability - markers - add info to link	2	2				
usability - markers - add info without picture	1	1				
usability - markers - color of markers	2			1		1
usability - markers - explain link marker and picture	1	1				
usability - markers - making groups	2		1	1		
usability - markers - missing links	1		1			

opmerkingen	Total Of Name	01 Intro	02 Stud. analysis	03 Guided analysis	04 Instruction	05 Search in database	06 Evaluation
01 Carmen	7			7			
02 Jilles	2		2				
03 Mikki	5		5				
05 Si	1		1				

The use of the markers led to a large number of problems. They were noted especially during the first number of sessions, but also during the guided analysis.

Findings

Usability – color of texts

Texts are not always readable: default text should have a contrasting color.	01 Carmen
Usability – markers - add info to link	
Students should make explicit why they select a certain keyword	01 Carmen
Without adding texts to links it will not always be clear to other readers why the link is	01 Carmen
Usability – markers – add info without picture	
It should be possible to add information or a marker without adding a picture to it.	01 Carmen
Usability – color of markers	
Default color of markers is light gray: marker gets easily lost or overlooked in picture	03 Mikki
You cannot change color of markers afterwards	05 Sigrun
Usability – markers – explain link marker and picture	
It is not obligatory (possible) to add information to a link between a marker and a picture.	01 Carmen
Usability – markers – making groups	
When starting: not clear that a group should be made	02 Jilles
Student wants to link picture to a group of markers. Is not possible.	03 Mikki
Usability – markers – missing links	
If the program gives feedback on missing links, the marker is selected. First it has to be deselected before the student can proceed.	02 Jilles
When two markers are on top of each other, you cannot access the lower marker	01 Carmen
Usability – markers – placing	
You cannot change the place of an arrow once it is put on the drawing.	03 Mikki
Usability – markers – size marker texts	
It is not clear how much text the information marker can store. It is also annoying that you cannot add new-lines and indents to make-up texts	01 Carmen
Usability – markers – toolkit	

When a form is used as a marker (rectangle) it is shown in toolkit as a massive block. Blocks view. In printview it is semi-transparent. Is very confusing.	03 Mikki

Usability details: printview

Name	Total Of Name	1	2	3	4	5
usability - printview	6	2	4			
usability - printview - keywords on top	1					1
usability - printview - information	1		1			

opmerkingen	Total Of Name	01 Intro	02 Stud. analysis	03 Guided analysis	04 Instruction	05 Search in database	06 Evaluation
01 Carmen	2			2			
02 Jilles	5		3	2			
05 Si	1		1				

Findings

Usability – printview	
In printview the texts are not shown.	01 Carmen
Pictures are shown in printview in order in which they were added. When explaining for example routing it should be possible to influence the order in which pictures are.	02 Jilles
The numbers are missing in printview.	02 Jilles
Usability – printview – keywords on top	
Relation between keywords on top of printview and the rest of page is not clear.	05 Sigrun
Usability – printview – information	
You cannot read associated text when looking at a picture associated with a marker (in printview). Text should have contrasting color.	02 Jilles

Usability details: search

Name	Total Of Name	1	2	3	4	5
usability - search	2					2
usability - search - feedback on search	1			1		
usability - search - feedback selected keywords	1	1				

opmerkingen	Total Of Name	01 Intro	02 Stud. analysis	03 Guided analysis	04 Instruction	05 Search in database	06 Evaluation
01 Carmen	1			1			
03 Mikki	1						1
05 Si	2		2				

Findings

Usability – search

Search: feedback on which keywords is seached and how they are combined	05 Sigrun
Possibility to search on a specific architect or name of a building.	05 Sigrun

Usability – search – feedback on search

Not clear in feedback search which keywords are used to search and how they are combined (and, or).	03 Mikki

Usability – search – feedback selected keywords

You cannot see which keywords are selected: confusing when a keyword is selected in the list while it is not visible anymore. It should be clear which keywords are selected.	01 Carmen

Usability: other

Name	Total Of Name	1	2	3	4	5
bug	4		3		1	
usability - cookie	1	1				
usability - full screen	2		2			
usability - info about building	1		1			
usability - shortcuts	1					1

opmerkingen	Total Of Name	01 Intro	02 Stud. analysis	03 Guided analysis	04 Instruction	05 Search in database	06 Evaluation
01 Carmen	1			1			
02 Jilles	6		4	1			1
04 Rosie	1						1
05 Si	1					1	

A number of bugs were detected/reported in the second session: often bugs do not appear during testing.

Findings

Bug

When you add a subjective keyword and you don't save it explicitly (press enter) it gets lost.	02 Jilles

First time students puts a lot of effort in subjective description. When information is lost, next time less effort in description: less quality	02 Jilles
When clicking on markers the information should popup.	02 Jilles
Usability – cookie	
The cookie expires after some while. Is confusing to student: what happens, do I lose work, what should I do. Should be better feedback.	01 Carmen
Usability – full screen	
A scroll bar is necessary next to keywords: now only in full screen mode subjective keywords can be added	02 Jilles
Idea of full-screen: not all students know how it works.	02 Jilles
Usability – info about building	
You cannot see which building is analyzed in DAN (not in plannenmap, not in printview)	02 Jilles
Usability – shortcuts	
Use standard shortcuts if possible for navigation: for example alt + and - for zooming	05 Sigrun

Sessions

Session	*Date*	*Start*	*Videofilename*	*Tester*	*Remarks*
1	12/2/2003	10:30:00	infobase 1 021203 1100.avi	01 Carmen	
2	12/3/2003	9:00:00	infobase 2 031203 0900.avi	02 Jilles	
3	12/3/2003	2:30:00 PM	infobase 3 031203 1500.avi	03 Mikki	
4	12/4/2003	9:30:00	infobase 4 041203 0930.avi	04 Rosie	
5	12/4/2003	1:00:00 PM	infobase 5 041203 1300.avi	05 Sigrun	

D

THE SYSTEM ARCHITECTURE OF THE FINAL DESIGNMAP PROTOTYPE IMPLEMENTATION

This appendix contains the data structure of the final prototype implementation of the DesignMap environment (see section 5.4). This structure was developed in UML, and the code was developed in Java.

Structure of the DesignMap environment

A UML model has been created to describe the different classes needed in DesignMap. UML employs a hierarchical approach that involves a number of layers. The top layer is shown in Figure D.1, where five classes are given, each of which contains a number of internal layered classes.

The DesignMap class, shown to the left of the figure, is the main class of the environment that gets initialized when it starts up, and eventually initializes all other classes in the DesignMap environment. The GUI is the graphical user interface class that contains all other components of DesignMap, and allows easy interaction between the user and the tool. In addition to the toolbar and the status bar, the GUI features the two main ways the environment employs to represent the design process: the KeywordTree component and the TouchGraph component. As their name properly suggest, the KeywordTree component models the tree browser in the environment, while the TouchGraph component models the TouchGraph browser. The thumbnail panel makes part of the GUI class itself. The fifth and final top-level class is the DesignMapData, which is where an internal representation is stored of the data structure related to the design process. This internal representation is a copy that each DesignMap client has of the one present in the common database all clients are supposed to connect to in the software environment (Figure D.2). In the following, the internals of the DesignMapData is discussed in more detail.

The data stored by the DesignMapData class represents a copy of the design data present in the database side of the application. This data is present in all active DesignMap clients connected to the network. Therefore, it is very important for these data structures in each client to stay up-to-date with the changes that take place in the database, and in other clients. In order to achieve this, the internal structure of the DesignMapData contains 8 classes, divided into the following two main groups (Figure D.2).

- Data storage classes
- Data communication classes

The data storage classes contain the actual DesignMap data as represented in the database. There are four data storage classes: the Keywords, the KeyKeyRelations, the Documents and the KeyDocRelations. The Keywords class is coded as a hash table of the keywords the designers use to describe their ideas. These keywords are related to other keywords by relationships described in the KeyKeyRelations vector class. A similar class representation is used to store the documents data, where the Documents class contains the actual documents stored by the designers, while the KeyDocRelations class contains the relationships each document has with the keywords.

The data communication classes are responsible of keeping the data within each client synchronized with that represented in the GUI component of the application, with that contained in other clients, as well as with that contained in the database. Four classes ensure this synchronization: the MainCommunication, the AppComm, the ClientComm, and the DBComm (Figure D.2). The MainCommunication is the central class that regulates the different updating activities that need to take place, depending on the origin of the modification issued to the application. The AppComm is the class that regulates the modifications that take place in application itself as shown in the tree browser, the TouchGraph browser and the thumbnail panel. Any modifications made by the user through the GUI are propagated by the AppComm to the DesignMapData, to the database, and to all other clients. In the same way, any modifications made by other users to the database are signaled by the MainCommunication, and propagated by the AppComm to the GUI of each other client. The DBComm is the class responsible of updating the database with the changes, while the ClientComm is the class responsible of updating all other clients of the changes taking place locally. This way, all modifications taking place in any client is sure to be synchronized with all other clients and the database.

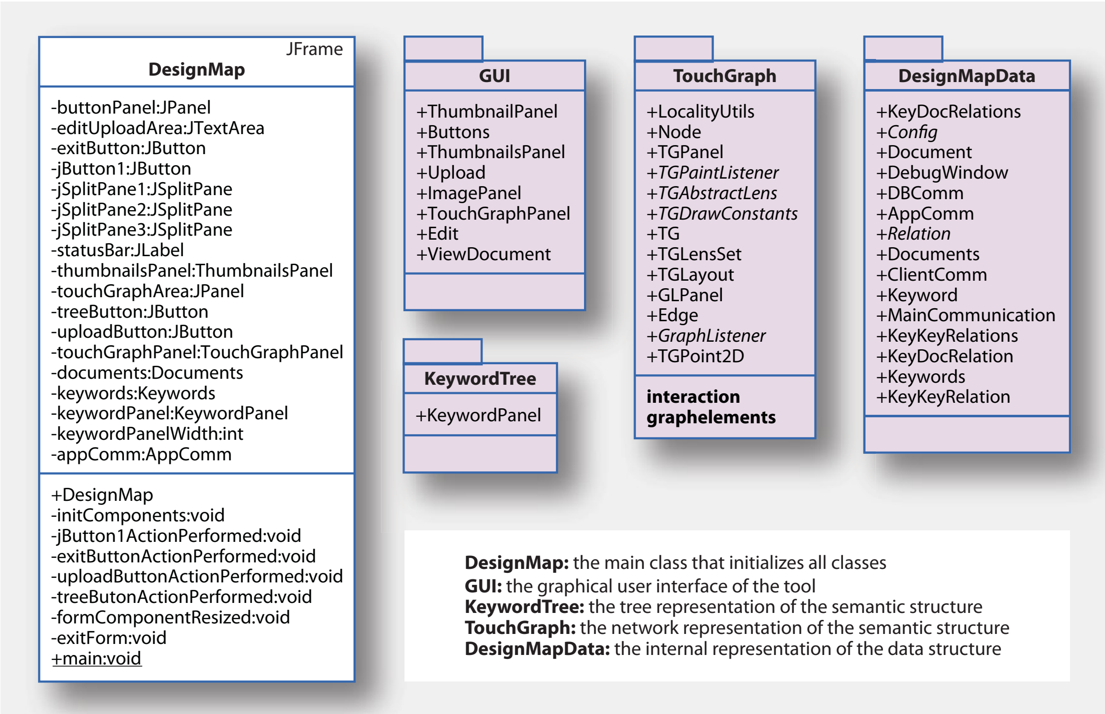


Figure D.1. The top level UML model of the DesignMap environment.

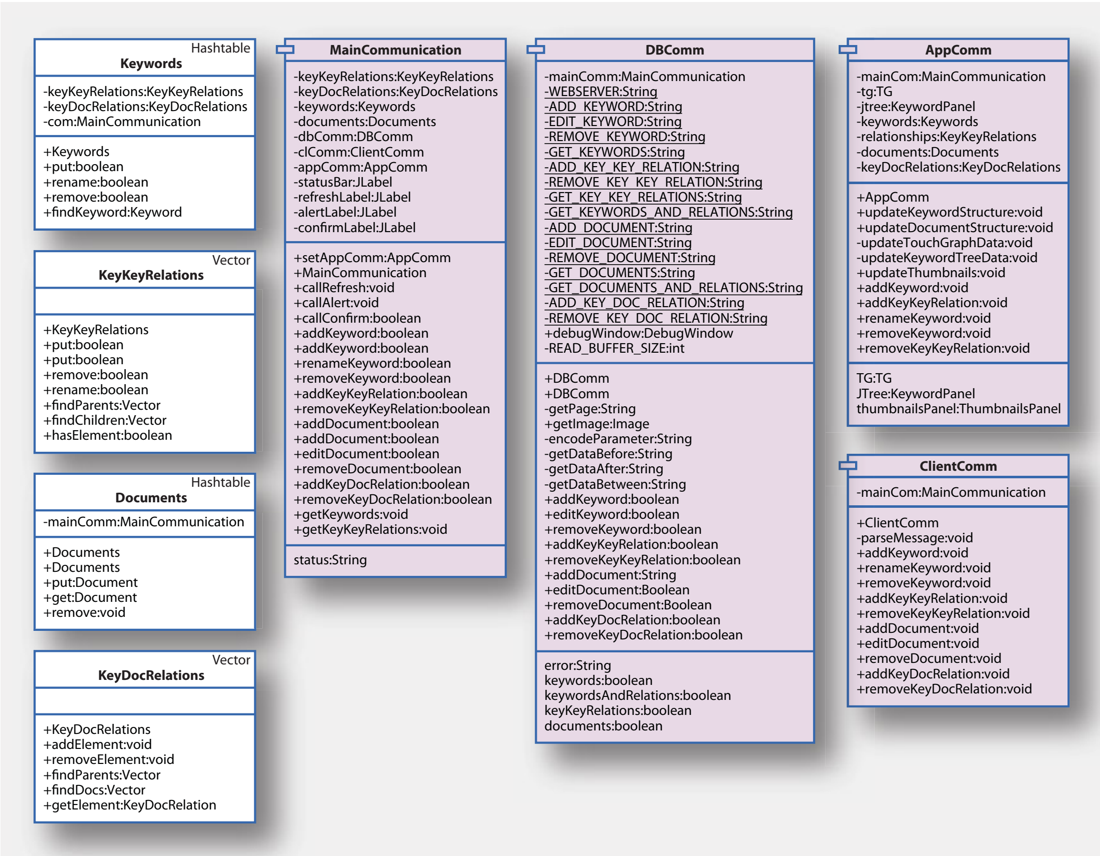


Figure D.2. The internal UML model of the DesignMapData class shown in Figure D.1.

SAMENVATTING

Dit onderzoek richt zich op de acquisitie, representatie, het delen en hergebruiken van ontwerp informatie en kennis in de conceptuele fase van het architectonisch ontwerp en is gericht op het creëren van gesitueerde digitale omgevingen waarin teams van ontwerpers communiceren en samenwerken met behulp van deze informatie en kennis.

Het belangrijkste product van de conceptuele architectonische ontwerpfase is een ontwerpconcept dat belooft met succes te worden ontwikkeld voor het gegeven ontwerpproject. Ter bevordering van de generatie van dit concept verzamelen ontwerpers informatie om kennis en inzichten over een ontwerp taak te verkrijgen, maar ook om inspiratie en creatieve ideeën op te doen. Precedenten - bekende voorbeelden van goede ontwerpoplossingen - fungeren als een gemeenschappelijke bron van kennis en inspiratie voor ontwerpers. Vele precedentbibliotheken bestaan waar (visuele) documenten worden verzameld in een depot (repository), meestal georganiseerd op basis van gemeenschappelijke categorieën zoals 'jaar van voltooiing' en 'architect'. Echter, in de conceptuele ontwerpfase zijn de ontwerpers over het algemeen niet bereid om specifieke vragen voor het ophalen van informatie te formuleren. Men kan geïnteresseerd zijn in het kijken naar alle documenten over een bepaald onderwerp, of gewoon het springen van link naar link, een bepaalde draad volgend. Dit vereist dat de informatiestructuur die de documenten relateert dicht genoeg is en tegelijk een organisatiestructuur bezit die een categorisatie van documenten toelaat die krachtiger is dan een eenvoudige verzameling van gemeenschappelijke categorieën. Bovendien, wanneer ontwerpers een organisatiestructuur definiëren voor de kennis en inzichten die ze krijgen van precedenten, versterkt dit hun ontwerpredenering proces. Dat wil zeggen, ontwerpers construeren een cognitief model van relevante verbindingen tussen het huidige probleem en het ontwerp logica aan de ene kant, en de kennis en concepten die ten grondslag liggen aan de precedenten opgeslagen in het depot aan de andere kant. Daarom stellen de digitale omgevingen waarop dit onderzoek zich richt hun gebruikers in staat om collectief, interactief en incrementeel een informatiestructuur te ontwikkelen die de informatie en kennis die in de omgeving verblijft organiseert.

Een gemeenschap van ontwerpers deelt vaak een gemeenschappelijke professionele taal, waarbij de woordenschat van deze taal een gedeeld begrip weergeeft. Deze taal is gevormd in de tijd en doorgegeven aan de nieuwe leden van de gemeenschap. Leden van een dergelijke ontwerpgemeenschap die samenwerken aan een gemeenschappelijk doel (meestal een project) vormen een praktijkgemeenschap (community of practice). Leden van een praktijkgemeenschap opereren zowel door registratie van gemeenschappelijke kennis in documenten en door actief deel te nemen aan sociale processen om deze vastgelegde kennis persoonlijk te contextualiseren. Deze activiteiten zijn zowel het middel als het resultaat van een architecturale praktijkgemeenschap. Door het uitvoeren van beide activiteiten zijn leden van een architecturale praktijkgemeenschap het gezamenlijk eens over de waarde van deze kennis en informatie. Dit wordt aangeduid als correspondentie, dat wil zeggen, communicatie met het doel een akkoord te bereiken. Correspondentie zorgt ervoor dat de makers van de informatie en kennis ook de gebruikers zijn. Correspondentie is de sleutel tot het ontstaan van een dichte en zeer intergerelateerde informatiestructuur die de basis vormt van digitale informatie-omgevingen voor de conceptuele fase van het ontwerp. Dergelijke informatiestructuren worden aangeduid als 'complexe informatiestructuren' in dit onderzoek.

Een complexe informatiestructuur is samengesteld uit informatie-entiteiten en hun relaties, geëtiketteerd met bepaalde ontwerpconcepten. Deze ontwerpconcepten zijn zelf verbonden door middel van semantische relaties die een semantische structuur vormen. Deze semantische structuur fungeert als de organisatorische ruggengraat van een complexe informatiestructuur. Elementen (concepten en relaties) van deze semantische structuur worden geassocieerd met informatie entiteiten (documenten) en beschrijven deze. Een complexe informatiestructuur die is gecreëerd door een praktijkgemeenschap door middel van informatie en sociale processen heeft kenmerken van een 'complex adaptief systeem', waarbij de structuur van het systeem niet-hiërarchisch is, de interacties niet vooraf gedefinieerd zijn, en de toestand van het systeem onvoorspelbaar is.

In dit onderzoek is de gefundeerde theorie (grounded theory) aangenomen als onderzoeksmethodologie om een op context gebaseerde en iteratieve benadering van het onderzoeksdomein en haar vraagstukken te ontwikkelen. In deze context is de onderzoeksvraag op iteratieve wijze geformuleerd als volgt: Hoe kunnen architecturale praktijkgemeenschappen corresponderen over ontwerpkennis en informatie tijdens de conceptuele fase van het ontwerp? Onderzoekscasussen hebben empirische en kwalitatieve gegevens opgeleverd om de theorie te funderen en iteratief te formuleren. Deze onderzoeksvraag is aangepakt door een studie van relevante literatuur, theorieën, methoden en technieken, en heeft geleid tot de ontwikkeling van een computationeel kader genaamd de Architectural Information Map (ArcIMap; Architectonische Informatie Kaart). Complexe informatiestructuren vormen de basis van ArcIMap. Het doel van ArcIMap is het definiëren van een structuur voor het ontwerp en de creatie van digitale toepassingen ter ondersteuning van ontwerpers in de conceptuele ontwerpfase door het definiëren van het representatieve kader voor de totstandbrenging van een geïntegreerde informatiestructuur van componenten, relaties en metadata uit een collectie van ontwerpdocumenten en de kennis die in deze documenten aanwezig is. Het kader kan vervolgens worden geïmplementeerd voor verschillende doeleinden, domeinen, contexten, of architecturale organen.

ArcIMap is zowel een methode als een model. De methode definieert sociale en informatieprocessen met het oog op het maken van complexe informatiestructuren die ten grondslag liggen aan complexe adaptieve systemen. Het model fungeert als een structuur voor het ontwerp van complexe informatiestructuren. De technieken en technologieën die ingekapseld zijn in het model staan de implementatie van toepassingen van ArcIMap in diverse educatieve en praktische contexten toe. Een toepassing van ArcIMap moet geworteld zijn in de gebruikscontext, dus, een studie van de sociale en werkprocessen van de gebruikers en de organisatorische structuur van de context waarin het zal worden gebruikt moeten worden bestudeerd in de ontwerpfase van de applicatie. Omgevingen die gebruikt zullen worden in een onderwijs context hebben andere eisen dan deze die in de praktijk zullen gebruikt worden, omdat ervaren ontwerpers andere behoeften hebben dan nieuwelingen.

Vier prototype toepassingen van ArcIMap zijn ontwikkeld, gesitueerd en geëvalueerd in verschillende architectonische onderwijs en praktijk contexten. Deze toepassingen en de evaluatie daarvan hebben waardevolle feedback geleverd aan de theorievorming en aan de iteratieve definitie van ArcIMap.

De eerste toepassing is een analyse presentatie tool dat drie Ottomaanse moskeeën gebruikt als casussen, dat het gezamenlijke representatieve kader onderzoekt en valideert, en dat de begrippen interactie en associatief bladeren in een toepassing van ArcIMap evalueert. De tweede toepassing, Blob Inventarisatie Project (BLIP), is een precedentbibliotheek ontworpen voor het modelleren van kennis die is voortgekomen uit digitaal ontwerpen, engineering en de productie van vrije-geometrische-vorm gebouwen en is gebruikt geweest in het 3e semester van het M Sc. architectuur onderwijs. BLIP onderzoekt en evalueert de gebruikersinterface en de interactie aspecten van ArcIMap. De derde toepassing, Design Analysis Network (DAN; Ontwerp Analyse Netwerk), is een informatiesysteem dat geïmplementeerd is als een educatieve architectonische analyse omgeving en gebruikt is in een 2de jaars (B.Sc.) ontwerpatelier. DAN onderzoekt en beoordeelt alle componenten van ArcIMap, maar vooral de inbedding in een context. De vierde toepassing, DesignMap, is een flexibel en uitbreidbaar content management systeem bestemd om te worden gebruikt in de vroege stadia van het ontwerpen, is gericht op kleine en middelgrote architectenbureaus, en werd gebruikt en geëvalueerd op het architectenbureau Mecanoo in Delft. DesignMap onderzoekt, implementeert en evalueert ArcIMap binnen de context van de architecturale praktijk.

De bruikbare resultaten van dit onderzoek zijn de ArcIMap methode en model, en de vier prototype applicaties. Bovendien is Keyset, zoals het is ontwikkeld en gebruikt in DAN, met succes gebruikt door duizenden studenten van de Faculteit Bouwkunde, TU Delft, en aan de Faculteit der Sociale Wetenschappen, Universiteit van Utrecht, sinds 2003, waaruit het succes van het onderzoek blijkt.

ABOUT THE AUTHOR

Bige Tunçer received a Bachelor degree in Architecture from Middle East Technical University, Ankara, Turkey in 1993, and a Master of Science degree in Computational Design from Carnegie Mellon University, Pittsburgh, USA in 1996. Later she worked for three years as a junior faculty member at the Chair for Architecture and CAAD, Swiss Federal Institute of Technology in Zurich, Switzerland. She is currently assistant professor at the Design Informatics chair and a liaison nestor at the Delft School of Design at the Faculty of Architecture, Delft University of Technology.

Her research interests include information and knowledge modeling and visualization, collaborative design, e-learning, and design exploration and generative design. She has been principal investigator and participant of a number of funded research projects. Her research has been published in books, journals and conference proceedings. She regularly leads scientific workshops and gives lectures internationally. She is a member of scientific committees of various journals and conferences. She has organized two international peer-reviewed conferences and chaired one of these.

She has conceived and taught numerous courses since 1996. Currently, she teaches digital design studios to B.Sc. and M.Sc. students, and digital fabrication and design theory to M Sc. students. She is advisor to a number of M Sc. graduation projects. She has organized and participated in three interdisciplinary virtual design studios, for one of which she was granted an AIA Education Honor Award in 2008.

She regularly takes part in academic committees at the Faculty of Architecture. She was the assistant research coordinator of the Department of Building Technology between 2006 and 2008. During this time she organized peer reviews for the research programs of the department and developed research strategies. As part of her work at the Delft School of Design, in 2007-2008, she organized the Architectural Engineering: Performance, Geometry, Materials program consisting of lectures and workshops, which assembled some of the world's leading academicians and practitioners.

PUBLICATIONS OF THE AUTHOR

EDITED BOOKS AND JOURNAL ISSUES

Sariyildiz S. and Tunçer B. (Eds.) (2005). INNOVATION in ARCHITECTURE, ENGINEERING AND CONSTRUCTION, Volume 1 - Innovation in Design, Engineering and Construction, SOON, Rotterdam, The Netherlands.

Sariyildiz S. and Tunçer B. (Eds.) (2005). INNOVATION in ARCHITECTURE, ENGINEERING AND CONSTRUCTION, Volume 2 - INNOVATION IN TECHNOLOGY and Management in AEC, SOON, Rotterdam, The Netherlands.

Beheshti R. and Tunçer B. (Eds.) (2005). Supporting Design Environments, International Journal of Design Sciences and Technology (Special Issue) 12(1).

Tunçer B., Ozsariyildiz S S. and Sariyildiz S. (Eds.) (2003). E-Activities in Design and Design Education, Europia Productions, Paris.

Tunçer B., Ozsariyildiz S.S. and Sariyildiz S. (Eds.) (2003). E-Activities in Building Design and Construction, Europia Productions, Paris.

BOOK CHAPTERS

Tunçer B. (2009). Data exchange study, 751 Multiplayer Design Studio, 14-18.

Kocaturk T., Tunçer B. and Veltkamp M. (2004). BLIP: Blob Inventory Project, BLOB in the Faculty 2004 (eds. M. Eekhout et al.), Faculty of Architecture, Delft University of Technology, The Netherlands, 34-37.

Sariyildiz S., Stouffs R., Çiftçioglu Ö. and Tunçer B. (2002). Future ICT Developments, Ways to Study and Research (eds. T.M. de Jong and D J.M. van der Voordt), DUP Science, Delft, The Netherlands, 377-383.

REFEREED JOURNAL ARTICLES

Akar E., Öztürk E., Tunçer B. and Wiethoff M. (2004). Evaluation of a collaborative virtual learning environment, Education + Training 24(6+7), 343-352.

Stouffs R., Kooistra J. and Tunçer B. (2004). METADATA AS A MEANS FOR CORRESPONDENCE ON DIGITAL MEDIA, ITcon Vol. 9, Special Issue Digital Media Libraries, pp. 129-142. [www.itcon.org/2004/9]

Tunçer B., Stouffs R. and Sariyildiz I.S. (2002). Document decomposition by content as a means for structuring building project information, Construction Innovation 2(4), 229-248 [Invited paper].

Zreik K., Beheshti R., Stouffs R., Tuncer B. and Ozsariyildiz S.S. (2002). The building and construction eDocuments, International Journal of Design Sciences and Technology 10(1), 7-16.

Akin Ö., Cumming M., Shealey M. and Tunçer B. (1997). An electronic design assistance tool for case-based representation of designs, Automation in Construction 6, pp. 265-274 [Invited paper].

REFEREED CONFERENCE ARTICLES

Tunçer B., de Ruiter P. and Mulders S. (2008). An Experiment in Multidisciplinary Digital Design, Architecture in Computro [26th eCAADe Conference Proceedings] (pp. 659-666). Antwerp, Belgium.

Tunçer B. (2007). Modeling of knowledge for free form building processes. In H. Dashti (Ed.), Sources of Architectural Form: Theory and Practice. Kuwait: Kuwait University.

Tunçer B.,Sariyildiz S. and Stouffs R. (2006). Arcimap : knowledge modeling for conceptual building design, in H Rivard, E Miresco, & H Melhem (Eds.), Proceedings - Building on IT; Joint International Conference on Computing and Decision Making in Civil and Building Engineering (pp. 2015-2024). Montréal, Canada.

Tunçer B. and Sariyildiz S. (2006). Design analysis network. In V Bourdakis & D Charitos (Eds.), eCAADe 24 Proceedings of the 24th conference on education in computer aided architectural design in Europe : comminicating space(s) (pp. 206-214). Volos: University of Thessaly.

Tunçer B., Al-Ars Z., Akar E., Beintema J. and Sariyildiz S. (2006). Designmap: capturing design knowledge in architectural practice. In M Dulaimi (Ed.), Conference Proceedings of The Joint International Conference on Construction Culture, Innovation and Management (CCIM) (pp. 352-361). Dubai, UAE: BUiD.

Kooistra J., Hopstaken K., Stouffs R., Tunçer B., Janssen Groesbeek E. and Sjoer E. (2005). Keyset: Conceptual, technical, didactic and strategic qualities of a 'discourse browser', INNOVATION in ARCHITECTURE, ENGINEERING AND CONSTRUCTION, Volume 1 - Innovation in Design, Engineering and Construction (eds. S. Sariyildiz and B. Tunçer), Rotterdam, The Netherlands, 353-364.

Stouffs R., Kooistra J., Tunçer B. and Akar E. (2005). Design communication and collaboration, Design Research in the Netherlands 2005 (eds. H. Achten, K. Dorst, P.J. Stappers and B. de Vries), Eindhoven University of Technology, Eindhoven, The Netherlands, 85-94.

Stouffs R., Tunçer B., Janssen Groesbeek E., Kooistra J. and Sjoer E. (2005). The use of metadata as a means to improve the quality of design: results from education, CAADRIA 2005 (Proceedings of the 10th International Conference on Computer Aided Architectural Design Research in Asia), Volume 2 (ed. A. Bhatt), TVB School of Habitat Studies, New Delhi, India, 503-509.

Tunçer B., Ciftcioglu Ö., Sariyildiz S. and Cumming M. (2005). Intelligent design support, CAADRIA 2005 (Proceedings of the 10th International Conference on Computer Aided Architectural Design Research in Asia), Volume 2 (ed. A. Bhatt), TVB School of Habitat Studies, New Delhi, India, 436-446.

Kocatürk T. and Tunçer B. (2004). Modeling Collaborative Conceptual Design in the Digital Age, Third International Workshop on Construction Information Technology in Education (ed. A. Dikbas), ITU Press, Istanbul, Turkey, 149-163.

Kooistra J., Stouffs R. and Tunçer B. (2004). Metadata as means for improving the quality of design, Cybernetics and Systems 2004 (ed. R. Trappl), vol. 1, Austrian Society for Cybernetic Studies, Vienna, 108-113.

Sariyildiz S., Ciftcioglu O., Tunçer B. and Stouffs R. (2004). Knowledge of Cultural Analogy in Design and Design Education — Multiculturalism, Local Values in a Networked Design World - Added Value of Computer Aided Architectural Design, DUP Science, Delft, The Netherlands, 91-104.

Stouffs R., Tunçer B., Janssen Groesbeek E. and Kooistra J. (2004). Keyset: The Use of Metadata as a means to improve the quality of design, Third International Workshop on Construction Information Technology in Education (ed. A. Dikbas), ITU Press, Istanbul, Turkey, 79-94.

Stouffs R., Engeli M. and Tunçer B. (2004). Mediated discourse as a form of architectonic intervention, Local Values in a Networked Design World - Added Value of Computer Aided Architectural Design, DUP Science, Delft, The Netherlands, 197-206.

Tunçer B., Çiftçioglu Ö., Sariyildiz S. and Cumming M. (2004). Intelligent design support: an educational experiment in knowledge modelling, Third International Workshop on Construction Information Technology in Education (ed. A. Dikbas), ITU Press, Istanbul, Turkey, 29-46.

Akar E., Tunçer B., Attema J. and Stouffs R. (2003). Design and evaluation of a collaborative virtual space: InfoBase, E-Activities in Design and Design Education (eds. B. Tunçer, S.S. Özsariyildiz and S. Sariyildiz), Europia Productions, Paris, 3-12.

Akar E., Andriessen J.H.E., Attema J. and Tunçer B. (2003). Socio-technical evaluation of computer supported work and learning systems, Human Computer Interaction: Theory and Practice, Part II (eds. C. Stephanidis and J. Jacko), Lawrence Erlbaum Assoc., Mahwah, NJ, 834-838.

Kocaturk T., Veltkamp M. and Tunçer B. (2003). Exploration of interrelationships between digital design and production processes of free-form complex surfaces in a web-based database, Digital Design: Research and Practice, Proceedings of the 10th International Conference on Computer Aided Architectural Design Futures (eds. M-L. Chiu, J-Y. Tsou, T. Kvan, M. Morozumi and T-S. Jeng), Kluwer Academic, Dordrecht, The Netherlands, 445-454.

Kooistra J., Stouffs R. and Tunçer B. (2003). Metadata as a means for correspondence in design analysis, E-Activities in Design and Design Education (eds. B. Tunçer, S S. Özsariyildiz and S. Sariyildiz), Europia Productions, Paris, 19-28.

Sariyildiz S., Ciftcioglu O., Stouffs R., Tunçer B. and Ozsariyildiz S.S. (2003). Information, communication and knowledge technology in the building sector, E-Activities in Building Design and Construction (eds. B. Tunçer, S S. Özsariyildiz and S. Sariyildiz), Europia Productions, Paris, 185-190.

Sariyildiz I.S., Tunçer B., Ciftcioglu Ö. and Stouffs R. (2003). An intelligent cooperation environment for the building sector, why?, ISEC-02: System-based Vision for Strategic and Creative Design, Volume 2 (ed. F. Bontempi), A A. Balkema, Lisse, The Netherlands, 1559-1565.

Stouffs R., Tunçer B. and Sariyildiz I.S. (2003). InfoBase: a multimedia learning environment to support group work and discourse, ISEC-02: System-based Vision for Strategic and Creative Design, Volume 2 (ed. F. Bontempi), A A. Balkema, Lisse, The Netherlands, 1751-1756.

Zreik K., Stouffs R., Tunçer B., Ozsariyildiz S. and Beheshti R. (2003). Information modelling for improving communication in design and construction, Digital Design: Research and Practice, Proceedings of the 10th International Conference on Computer Aided Architectural Design Futures (eds. M-L. Chiu, J-Y. Tsou, T. Kvan, M. Morozumi and T-S. Jeng), Kluwer Academic, Dordrecht, The Netherlands, 63-72.

Sariyildiz S., Stouffs R. and Tunçer B. (2002). Vision on ICT developments for the building sector, ACADIA 2000: Eternity, Infinity and Virtuality in Architecture (eds. M.J. Clayton and G.P. Vasquez de Velasco), Association for Computer-Aided Design in Architecture, 11-18.

Stouffs R., Tunçer B., Venne R.F. and Sariyildiz I S. (2002). InfoBase: a multimedia learning environment to support group work, Construction Information Technology in Education (ed. D. Rebolj), International Council for Research and Innovation in Building and Construction, Rotterdam, The Netherlands, 75-82.

Stouffs R., Tunçer B. and Sariyildiz S. (2002). WoonWerf.nl revisited: the potential of web-based design communication with future clients, Design and Decision Support Systems in Urban Planning, Eindhoven University of Technology, Eindhoven, The Netherlands, 240-250.

Stouffs R., Tunçer B. and Sariyildiz S. (2002). Empowering individuals to design and build collaborative information spaces, Distributing Knowledge in Buildings (eds. K. Agger, P. Christiansson and R. Howard), vol.2, Aarhus School of Architecture and Centre for Integrated Design, Aarhus, Denmark, 133-140.

Stouffs R., Tunçer B. and Sariyildiz S. (2002). Examples of web-based custom design in Dutch residential developments, Distributing Knowledge in Buildings (eds. K. Agger, P. Christiansson and R. Howard), vol.1, Aarhus School of Architecture and Centre for Integrated Design, Aarhus, Denmark, 148-155.

Tunçer B., Stouffs R. and Sariyildiz S. (2002). Cooperating on Architectural Analyses, eCAADe 20 [design e-ducation] Connecting the Real and the Virtual (eds. K. Koszewski and S. Wrona), eCAADe and Warsaw University of Technology, Warsaw, 20-27.

Tunçer B., Stouffs R. and Sariyildiz I.S. (2002). A toolkit for modelling architectural analyses in a design studio context, Construction Information Technology in Education (ed. D. Rebolj), International Council for Research and Innovation in Building and Construction, Rotterdam, The Netherlands, 59-64.

Tunçer B., Stouffs R. and Sariyildiz S. (2002). Managing architectural analyses in a collaborative context, Common Ground: Design Research Society International Conference 2002 (eds. D. Durling and J. Shackleton), Staffordshire Univ. Press, Stoke on Trent, UK, 1120-1134.

Tunçer B. and Stouffs R. (2002). Modeling cooperative design analyses, Design and Decision Support Systems in Architecture, Eindhoven University of Technology, Eindhoven, The Netherlands, 364-377.

Zreik K., Stouffs R., Tunçer B., Ozsariyildiz S. and Beheshti M.R. (2002). An eDocument approach for improving communication in AEC projects, eWork and eBusiness in Architecture, Engineering and Construction (eds. Z. Turk and R. Scherer), A.A. Balkema, Lisse, The Netherlands, 335-341.

Sariyildiz I.S., Stouffs R. and Tunçer B. (2001). ICCT influence on spatial planning, design and construction process, Information and Communication Technology (ICT) in the Practice of Building and Civil Engineering, Association of Finnish Civil Engineers RIL, Helsinki, 137-142.

Stouffs R., Venne R.F., Sariyildiz S. and Tunçer B. (2001). Aspects and technologies of e-learning in an architectural context, Architectural Information Management (ed. H. Penttilä), eCAADe and Helsinki University of Technology, Helsinki, Finland, 358-363.

Stouffs R., Sariyildiz S. and Tunçer B. (2001). ICT influences on design creativity, Digital Creativity: Crossing the Border (eds. R. Shaw, and J. McKay), The Glasgow School of Art Press, Glasgow, UK, 383-386.

Stouffs R., Tunçer B. and Sariyildiz S. (2001). The customer is king: web-based custom design in residential developments, CAADRIA 2001 (eds. J S. Gero, S. Chase, and M. Rosenman), Key Centre of Design Computing and Cognition, University of Sydney, Sydney, 149-157.

Tunçer B., Stouffs R. and Sariyildiz S. (2001). Integrating architectural abstractions, Reinventing the Discourse (ed. W. Jabi), Association for Computer-Aided Design in Architecture, Buffalo, N.Y., 110-121.

Tunçer B., Stouffs R. and Sariyildiz S. (2001). Rich information structures, Architectural Information Management (ed. H. Penttilä), eCAADe and Helsinki University of Technology, Helsinki, Finland, 30-35.

Tunçer B., Stouffs R. and Sariyildiz S. (2001). Flexibility and effectiveness in building project document management, Perspectives on Innovation in Architecture, Engineering and Construction (eds. C.J. Anumba, C. Egbu, and A. Thorpe), Centre for Innovative Construction Engineering (CICE), Loughborough University, UK, 729-737.

Tunçer B., Stouffs R. and Sariyildiz S. (2001). (Re)presentation of architectural analyses: two prototype applications, Computer Aided Architectural Design Futures 2001 (eds. B. de Vries, J. van Leeuwen, and H. Achten), Kluwer Academic, Dordrecht, The Netherlands, 495-505.

Tunçer B., Stouffs R. and Sariyildiz S. (2001). Types and documents: structuring building project information, Proceedings of the CIB-W78 International Conference: IT in Construction in Africa 2001, CSIR, Pretoria, South Africa, 20.1-20.13.

Tunçer B., Stouffs R. and Sariyildiz S. (2001). Document decomposition by content as a means for structuring building project information, Advances in Building Informatics (ed. R. Beheshti), Europia Productions, Paris, 209-221.

Tunçer B., Stouffs R. and Sariyildiz S. (2001). Facilitating the complexity of architectural analyses, AVOCAAD Third International Conference (eds. K. Nys, T. Provoost, J. Verbeke, and J. Verleye), Hogeschool voor Wetenschap en Kunst, Brussels, 114-121.

Sariyildiz I.S., Çiftçioglu Ö., Stouffs R. and Tunçer B. (2000). Future developments of ICT in the building sector, Construction Information Technology 2000 (ed. G. Gudnason), Icelandic Building Research Institute, Reykjavik, Iceland, vol. 2, 790-800.

Tunçer B. and Stouffs R. (2000). A representational framework for architectural analysis, Constructing the Digital Space: SIGRADI 2000 (eds. J. Ripper Kos, A. Pessoa Borde, and D. Rodriguez Barros), PROURB, Federal University of Rio de Janeiro, Rio de Janeiro, 206-208.

Tunçer B. and Stouffs R. (2000). Modeling building project information, Construction Information Technology 2000 (ed. G. Gudnason), Icelandic Building Research Institute, Reykjavik, Iceland, vol. 2, 937-947.

Tunçer B., Stouffs R. and Sariyildiz I.S. (2000). Collaborative information structures: educational and research experiences, COOP 2000 workshop proceedings: Analysing and Modelling Collective Design, INRIA, Rocquencourt, France, 20-28.

Papanikolaou M. and Tunçer B. (1999). The Fake.Space Experience - Exploring New Spaces, Architectural Computing: from Turing to 2000 (eds. A. Brown, M. Knight, and P. Berridge), eCAADe and The University of Liverpool, Liverpool, UK, 395-402.

Tunçer B. and Stouffs R. (1999). Computational richness in the representation of architectural languages, Architectural Computing: from Turing to 2000 (eds. A. Brown, M. Knight, and P. Berridge), eCAADe and The University of Liverpool, Liverpool, UK, 603-610.

Stouffs R., Kurmann D., Tunçer B., Mieusset K.H. and Stäger B. (1998) An information architecture for the virtual AEC company, Product and Process Modelling in the Building Industry (ed. R. Amor), Building Research Establishment, Watford, UK, 479-486.

Stouffs R., Tunçer B. and Schmitt G. (1998) Supports for information and communication in a collaborative building project, Artificial Intelligence in Design 98 (eds. J. Gero, and F. Sudweeks), Kluwer Academic, Dordrecht, The Netherlands, 601-617.

Akin Ö., Cumming M., Shealey M. and Tunçer B. (1996). An electronic design assistance tool for case-based representation of designs, Design Computation: Collaboration, Reasoning and Pedagogy, ACADIA Conference, Tucson, Arizona, USA, 123-132.

NON-REFEREED BOOKS, ARTICLES AND REPORTS

Eekhout M. and Tunçer B. (Eds.)(2007). Building Technology Research Programs Peer Review Evaluation, January-April 2007, Department of Building Technology, Faculty of Architecture, Delft University of Technology.

Eekhout M. and Tunçer B. (Eds.)(2007). Building Technology Research Projects, 31 December 2006, Department of Building Technology, Faculty of Architecture, Delft University of Technology.

Tunçer B., Borgart A. and Turrin M. (2006). Proceedings of Colloquium TU Delft Building Technology - IUAV Venice Architectural Construction, 26-27/6/2006, Faculty of Architecture, Delft University of Technology.

Borgart A., Kocaturk T., Tunçer B. and Veltkamp M. (Eds.)(2006). Proceedings of the First Free Form Design Colloquium, IASS and Faculty of Architecture, Delft University of Technology.

Beheshti R. and Tunçer B. (2005). Editorial: Information, communication and knowledge technologies supporting building design and construction processes, International Journal of Design Sciences and Technology 12(1), 7-12.

Tunçer B. and Kocaturk T. (2005). Women in Academia – "The Glass Ceiling", B-Nieuws nr. 17, volume 2004-2005.

Tunçer B. and Veltkamp M. (2004). The Bouwkunde Ph.D. Afternoon, Het Proefschrift: Journal for P.Hd. Students in Delft Autumn 2004, 10-12.

Stouffs R. and Tunçer B. (2000). Database ontwerp ter ondersteuning van Woonwerf.nl, Technical report, Chair for Technical Design and Informatics, Faculty of Architecture, Delft University of Technology, June 2000.

Stouffs R. and Tunçer B. (1997). Information architecture: presentation and representation, Technical report, Chair of Architecture and CAAD, Swiss Federal Institute of Technology Zurich, April 1997.