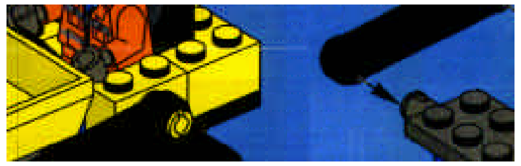


The pursuit of efficiency, profitability and improved quality has led up to increased interest in process thinking. By describing how the business is operated, i.e. what the processes look like, it is possible to improve and communicate a new way of working. The business can be graphically depicted and an intranet can distribute a picture of the business throughout the company.

Models of nouns are often interpreted without hindrances. Models of verbs are of a complete different kind. A course of event, a process, can be frozen as a snapshot, but how can these frozen pictures of changing states be made understandable? Developing a business by using process models needs *usable process models*. This licentiate thesis describes how process models presented in a computer environment can be made usable to practitioners of the construction industry.

VETENSKAP OCH KONST
KUNGL TEKNISKA HÖGSKOLAN

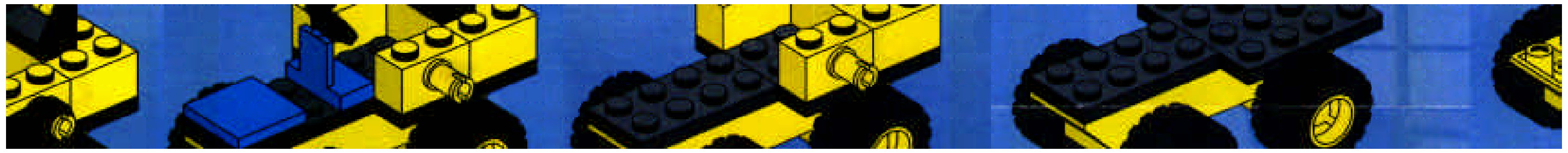
Department of Industrial Economics and Management
Construction Management and Economics

Stockholm, September 2001

Rikard Berg von Linde
Making Process Models Usable

Rikard Berg von Linde

Making Process Models Usable



Rikard Berg von Linde

Making Process Models Usable

Kungl Tekniska Högskolan

Department of Industrial Economics and Management
Construction Management and Economics

Stockholm, September 2001

© Rikard Berg von Linde 2001
Royal Institute of Technology, KTH
Department of Industrial Management and Economics
Construction Management and Economics
SE-100 44 Stockholm, Sweden

Printed by Högskoletryckeriet, KTH

ISBN 91-7283-153-7

Abstract

Process models have several fields of application. The research community of construction IT has used process modelling methodologies for several years to analyse and share information. The construction industry has applied process modelling, among other things, to better understand current business, to improve or innovate business and to create information systems that support business. Models of nouns are often interpreted without hindrances.

Models of verbs are of a complete different kind. A course of event, a process, can be frozen as a snapshot, but how can these frozen pictures of changing states be made understandable? Developing a business by using process models needs *usable process models*. This licentiate thesis describes how process models presented in a computer environment can be made usable to practitioners of the construction industry.

A structure of concepts is developed that describes interactive environments for process models. The concepts are of two different types: objects and actions. Objects are components that carry the information, and they are not examined in this research. The other type of concept, actions, describes activities that can be performed on objects in a process information system. Twelve different actions are identified in the thesis: overview, zoom, filter, details-on-demand, decompose, relate, history, extract, browse, search, compare and find patterns. For each action a widget (a component of the user interface) is presented that supports the action.

Based on the concepts developed, a prototype is developed. An authentic model built according to the IDEF0 process modelling method is displayed in the prototype. Finally, a usability study is performed to gain knowledge about the concepts and their implementation in the prototype.

The study shows that the prototype is, on the whole, making the fairly complicated IDEF0 model usable to the users of the study. The implementation itself also showed that the concepts have a practical value.

A secondary product of this research is the methodology that was used. It is presented in a way that, it is hoped, will inspire other researchers in the field of construction IT to more often conduct qualitative research and especially user and usability studies.

Preface

Some decisions are not as well considered as others. I was asked if I wanted to start as a PhD student in a construction IT research project at KTH. The problem was that I only had until the day after to make up my mind. The next day, Håkan Blom, CEO of Tyréns Byggkonsult AB, went to a steering group meeting of IT Bygg och Fastighet 2002 and announced that he had a candidate for the unfilled post in the MoPo project. The Tyréns' Foundation was furthermore willing to finance the post. This was how it all started and I do not regret that I accepted the challenge.

I had worked at Tyréns for a year before I enrolled in the research studies. Development of the quality system had been my primary focus at Tyréns during that year. The system had a process structure; it described how the business was carried out in reality. I considered this a sympathetic feature and it caught my attention from the very beginning. The business itself, not a stiff-legged ISO standard, was in focus. The system needed some kind of information system that made it easily accessible. Without too much analysing I built a web application. The information was stored in a database and it was displayed dynamically to the users. A lot of thought was given to how to make the information easy to find and understand. Using the process model as a navigator was the natural solution to this problem. The process model had a hierarchical structure and the browser widget known from Windows' user interface was implemented. The system became appreciated for its evident simplicity and graphical plainness.

The research project that I joined was called MoPo, Models for the construction industry. Bo-Christer Björk, professor at KTH, coordinated the project, which engaged six persons at the beginning. My practical knowledge about process modelling caused an interesting meeting with advanced methodologies like IDEF0 and UML. It was immediately apparent that the gap between researchers' models and practitioners' models was enormous. Bring an IDEF0 model to practitioners of the construction industry and it will be a perfect failure. I asked myself what the problem really consisted of and conducted a study on IDEF0 that resulted in a conference paper. Information visualisation in combination with usability were the exciting knowledge domains that were going to help me find the answer to the question of how to make process models usable.

I not only want to show how to make process models usable, but I also want to describe it in a way that is usable to practitioners of the construction industry. The only reading instruction I will give is addressed to practitioners, who will probably profit most from chapter 1, Introduction, and chapter 4, Process visualisation.

Acknowledgements

Many have supported me, but it is the following who have contributed in a matchless way.

I appreciated Professor Bo-Christer Björk's intelligent supervision as much as the academic (and social) trips to Slovenia and Portugal. Professor Örjan Wikforss succeeded Bo-Christer Björk as my supervisor and he guided me in a gentle but enthusiastic way during my last year of research.

My employer, Tyréns, gave me a unique possibility by supporting me financially, and I expressly want to thank Håkan Blom for promoting the idea from the very beginning.

My wife, Caroline, and her knowledge about HCI and usability are probably the reason why this research got its direction. One cannot possibly get more tangible help and support from a life companion.

And, of course, not to forget all the test persons I have tormented during the user studies: I hope all of you enjoyed it as much as I did.

Contents

1 Introduction

Models have fascinated many of us since our very early years. Lego and Mechano continue to tickle our fancy long after we have grown up and should no longer be interested in toys. There is a magic about those toys that never disappears. They give us the means to create representations of reality and to build worlds of the imagination which would otherwise be impossible to express. Lego and Mechano could, in other words, be described as powerful modelling devices.

It is easy to understand that it is possible to describe or represent a physical object with a model. We make objects that we find interesting or important appear in the model. A noun in the reality or in an imagined reality is represented by another noun in the model.

Processes are about changes of state. A process model is thus a description of a development or a course of events. The process model contains the verbs that we think are characteristic of the process. But giving verbs visual representations is not as easy and intuitively achieved as is the case for nouns. A shortcut could be to follow the principle used by the comic papers, i.e., to cut the course of events into a number of snapshots. The reader compares the state in adjacent boxes and lets imagination and sense interpolate the process that is not shown in the gap between the boxes. Working in this way solves the problem too easily. It is what happens between the boxes of the comic strip that is really interesting to a process modeller. Or, in the case of the construction manager, the interesting process is what happens from drawing to ready-for-use building.

This thesis is about how visual representations of process models can be made usable to practitioners of the construction industry. It is about how to make verbs visible and how to create another ingenious tool that facilitates one of the most information-intensive businesses in the world.

1.1 Models in general

What a model really is has been thoroughly examined by several researchers. A somewhat simplified description is to say that a model is a representation of a view of reality. This definition is not sufficient since it limits models to dealing only with existing systems, the reality. Gustafsson et al. (1982) suggest the following definition:

> **A model** is a system that we have chosen because it depicts important properties of another system.

A system is "a regularly interacting or interdependent group of items forming a unified whole" according to Merriam Webster's dictionary (Merriam Webster 1995). The above definition of a model says that a model is a system, which is made up of several combined components. This system, the model, depicts chosen details of another "real system".

Models are used to describe or answer questions about a defined and limited part of the world or the problem under consideration. Models can be categorised in many ways. Some of the properties that can distinguish a model are, for example, its ability to deal with time (static vs. dynamic), state (discrete vs. continuous), randomness (deterministic vs. stochastic) and level of repression of details (abstractness vs. similarity) (Miller et al. 1999). In addition to these categories, Gustafsson (1982) has identified the following:

> **Formal vs. informal models.** Models described according to a formal language are formal. A procedure in a computer program written in a program language can be an example of a formal process model. Its visual properties are, however, clearly limited.
>
> **Internal vs. external models.** An external description deals only with input and output of a system, and the system is regarded as a "black box". This is a property that is more or less pronounced for all process models and it is strongly dependent on level of detail. At a low level of detail the process model becomes a more pronounced "black-box model". The more detailed the description

is, the more explicit the content of the box becomes and the less the "black-box model" remains.

Iconic models. An iconic model is a depiction in a scale other than that of the observed system. A model aeroplane for use in wind tunnel experiments, for example, is an iconic model. Figure 1.2 shows another example of an iconic model.

Analogical models. Analogical models describe processes that are hard to observe by comparing them to other, better-known processes. In electricity, Ohm's law is well known. When describing heat transfer through a construction element it is possible to refer to Ohm's law to explain the thermodynamic process. This is a way of using an analogical model.

Symbolic models. A symbolic model expresses characteristics of things by means of symbols. Verbal descriptions in written language and computer models written in computer language are symbolic models. Most process modelling methods use symbols to represent properties of the depicted system. For example, an arrow often symbolises a process or a sub-process.

Common to all models is that they have a purpose, a viewpoint and a level of detail (Keijer et al. 1994). It is the purpose of the model, the intended usage, which decides the most appropriate design. The validity of a model must also be evaluated starting from the purpose of the model. To clearly state the purpose is thus important.

Reality is perceived differently depending on who is observing it. The viewpoints of an architect and a tenant looking at the same miniature model of a house are not the same. The prospective tenant may try to figure out how to arrange the furniture in the rooms, while the architect may be thinking about handicap regulations, aesthetics or the connections between different rooms. It is evident that a model is more or less useful for different categories of users.

A model can try to handle several viewpoints, but if they are too disparate a choice must be made. The modelling technique chosen greatly affects the possibilities of handling more than one viewpoint. A normal photograph or a drawing offers two dimensions and only one viewpoint of the system. If the model is transferred into a computer environment, for example in the form of a virtual reality model, the number of dimensions will grow to three or even higher (see the discussion about dimensions on page 4). This will provide the opportunity to offer different viewpoints of the same model in the same representation. Letting the users walk through the model and observe everything that they regard as interesting, provided that precisely those details are reproduced, does this.

A model contains less information than does the system it depicts. The modeller has to decide what details to render and what details to omit. The purpose and the viewpoint must guide the modeller in this decision. Purpose, viewpoint and level of detail are concepts that together decide the design of the model. Figure 1.1 shows how those concepts relate to each other. The modelling method is, in many cases, a filter between what is desired to express and what is possible to express.

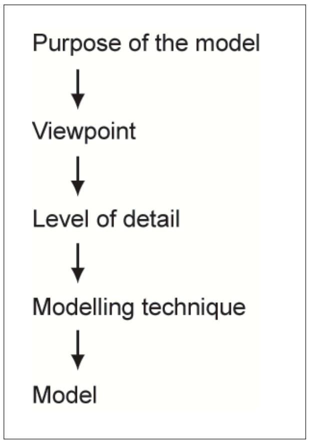


Figure 1.1 Relations between concepts that affect the design of a model. Gustafsson (1982) argues that the kind of system that is modelled also affects the representation, and this could be added to the figure.

The number of dimensions that the specific form of representation can handle extensively affects the visual expression and the interactive possibilities of representations. Miller et al. (1999) distinguish between six different dimensions:[1]

1D One-dimensional models are expressed as plain text (symbolic or formal models).

1.5D Hypertext has been added to the plain text, which makes it convenient to navigate between context and details.

2D Diagrams are examples of 2D models. Diagrams can also house 1D information, for example as comments and clarifications.

2.5D Ordinary diagrams are converted to hyper-diagrams. This makes it possible to quickly navigate between different views or levels of detail.

3D Virtual reality representations and physical objects (existing and non-existing) can be presented in the form of 3D models. Also, process models can be expressed as 3D models, for example as animations of a course of events.

3.5D By adding links between different locations in the 3D representation, the dimension becomes 3.5D. In a virtual reality model this is done by providing possibilities to interact with the model, i.e., making it possible for the user to control what to discover. An example of a virtual reality model that renders a process model is an animation of air movements in a refrigerating plant, in which the user can control different valves.

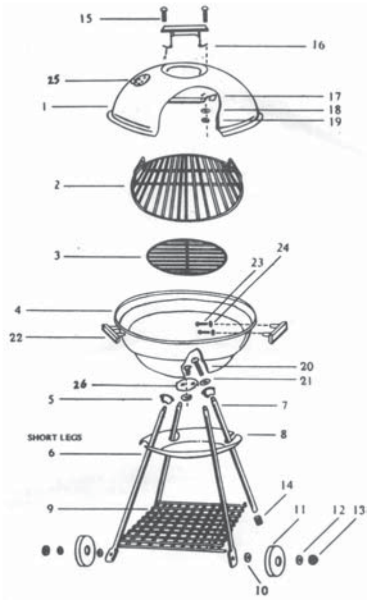


Figure 1.2 Example of an iconic model that shows how to assemble a barbecue.

The act of modelling, according to Gustafsson et al. (1982), is divided into two steps. At first the modeller tries to understand or conceptualise the real (or planned) system as a mental model. The next step is to express the mental model as an external model. The process of constructing a model is often as important as the result of the modelling because a solid understanding of the system is needed to develop a logically connected mental model. Modelling activities are often performed as group activities since the underlying mental models of a representation are difficult to transfer. Most effective transferral is done by active participation and common development of mental and external models.

1.2 Process models

The literal meaning of the word "process" is progress or course. Methods for managing processes have, according to Rentzhog (1998), been influenced by an engineering view, with roots back in Taylor's Scientific Management. Processes are regarded as standardised sets of activities that transform input to output. Rentzhog claims that social science has another view of processes that considers a process as the phenomenon of a change during a space of time. The structure and repetitiveness is less emphasised by this view. In this thesis both views are applicable. The process itself is not in focus, but the representation of the process, the process model, is. Most of the model categories discussed above are applicable to process models. An iconic process model may be hard to imagine, but consider a nicely illustrated users' manual, which in fact is an example of an iconic process model, see figure 1.2.

The working definition in this thesis is simply:[2]

> **A process** is a change of state in a system.

This definition indicates that a process has no value. Other definitions exist that give the process a value, for example, defining it as "a description of a set of related activities that, when correctly performed, will satisfy an explicit goal" (Eriksson and Penker 2000). A definition like this omits meaningless processes, which of course do exist. A product model describes nouns or objects of a system, and a process model describes verbs of a system. A common way to find out which objects are relevant to represent in a product model is to start by defining a process model. Designers of information systems work in a similar way. By defining the business processes, requirements on the information system can be captured. A data model can represent features of reality but also features of a computer system. The same is true for process models, e.g., a process model does not always have an obvious mapping to reality.

The scope for process models, however, is wider. Process models do not concern only requirements for information systems but also business and management systems. As said before, validity of a model is judged according to the model's purpose. If the purpose is no longer limited to systems development, the models must somehow be changed to be useful. Process management is not a new management discipline. Radical management concepts like business process reengineering (BPR) and process innovation have swept over the industry for several years. A survey of recent management literature shows that the word ”process” is used in many different ways. It is a risk that the meaning of the word becomes ambiguous. The international standard for quality management systems, ISO 9000, have given the process approach a key role in the standard. It is explained as "the application of a system of processes within an organisation, together with the identification and interactions of these processes, and their management" (ISO 9000:2000, 2000). According to the standard, processes must be defined, communicated, monitored and continually improved. Few of the different process management concepts that exist are precise and consistent with the use of the words "process" and "process model". A process model in some cases consists of vague textual descriptions and in other cases of complex graphics. The purpose of the model determines what kind of representation will be useful.

Figures 1.3 to 1.6 present examples of different process models, all of which have different purposes and in some respect have tried to adapt design to purpose. The size of the images makes it difficult to observe any details, but the purpose of the illustrations is only to exemplify some different ways in which a process model can be presented.

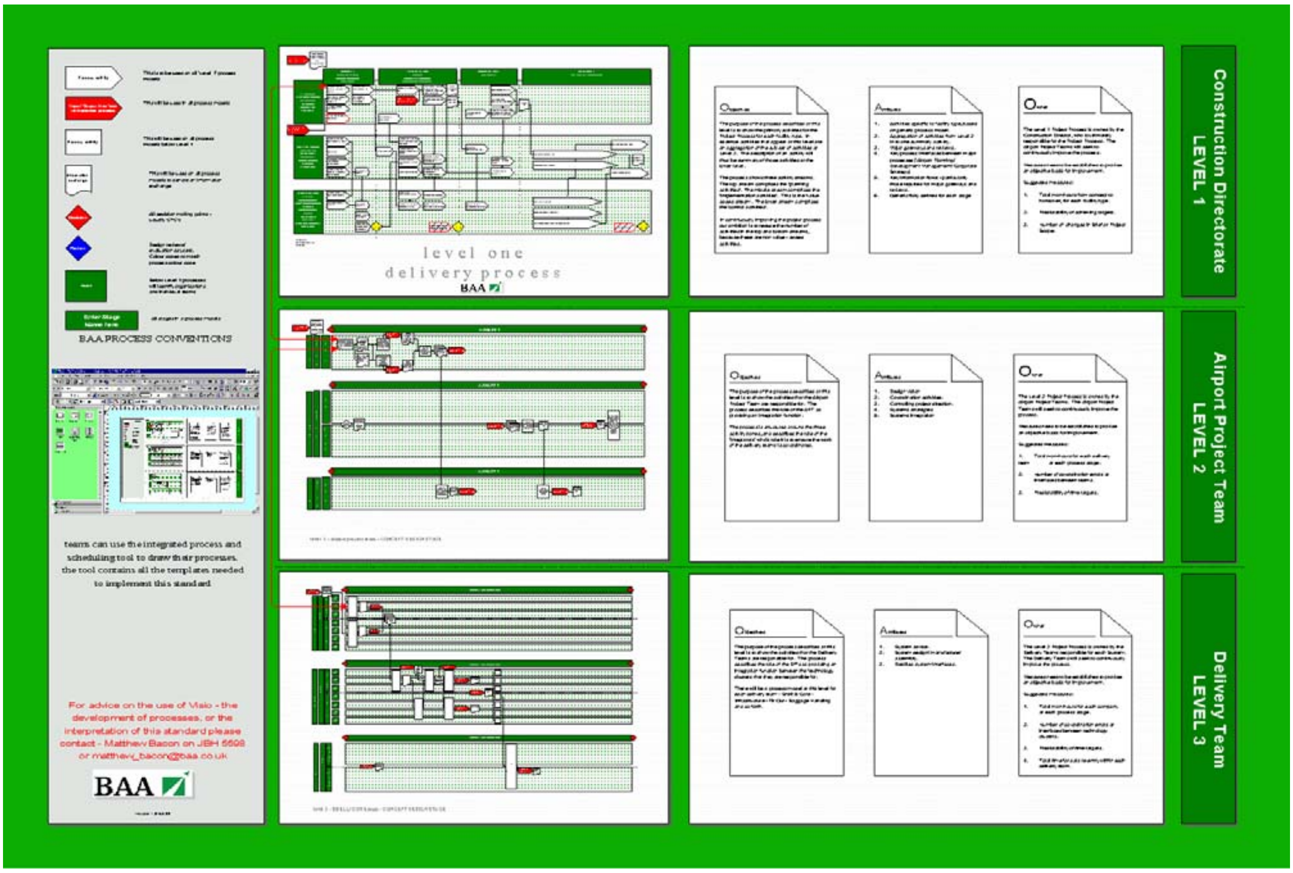


Figure 1.3 BAA (former British Airport Authority) is one of the UK's major developers of infrastructure and one of the construction industry's largest clients. In order to create a more effective construction process, detailed process models were developed defining how to run BAA's construction projects. Many different functions at an airport are affected when a construction project is performed, and all functions need to know their role in this complicated system. The process models were printed in colour on huge sheets of paper. The process model depicted in this figure is heavily reduced in scale.

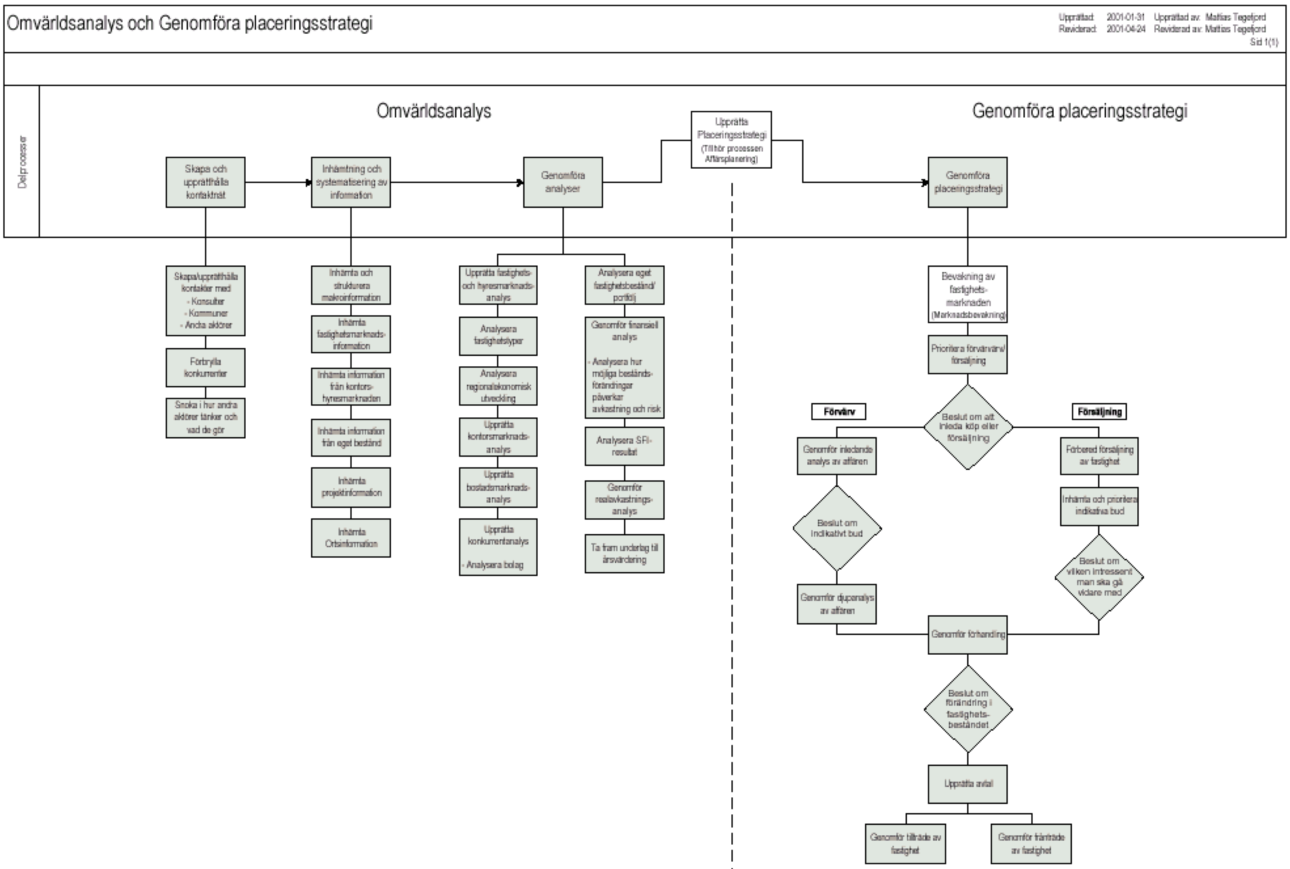


Figure 1.4 AP Fastigheter is one of Sweden's largest facility managers. All business processes of the company have been systematically described and mapped graphically as in this figure. This example shows the process of analysing conditions and making investments. People in the organisation have produced the process models (with guidance from consultants) and the process modelling as such has been considered as important as the resulting models. The process models have been used to define need of administrative aid (ready to use templates, forms, etc.) and to visually present working procedures on different levels in the company.

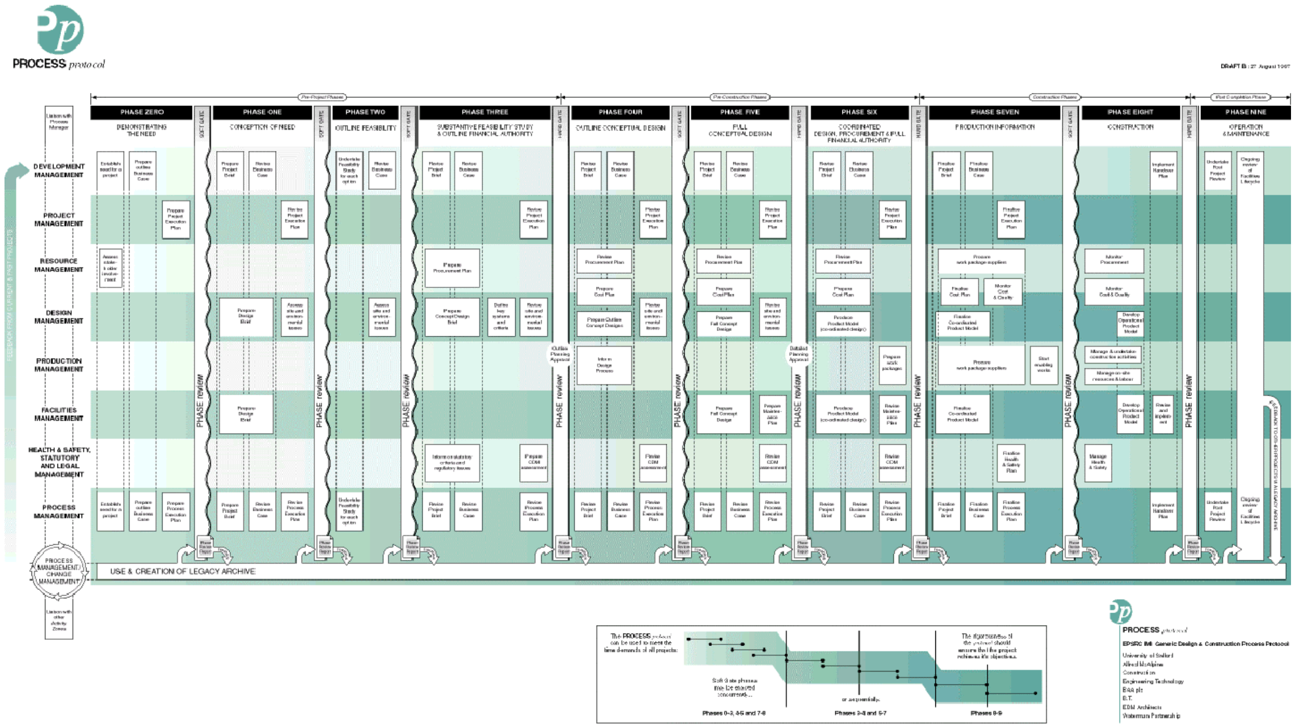


Figure 1.5 The Process Protocol is an extensive British research initiative with participants from both academia and the industry. It is a common set of definitions, documentations and procedures that are produced to help organisations within the construction industry to work together seamlessly (Kagiouglou et al. 1998). The Process Protocol breaks down the design and construction process into 10 phases. These 10 phases are grouped into 4 broad stages, namely Pre-Project, Pre-Construction, Construction and Post-construction. The 4 stages are outlined at the top of the map.

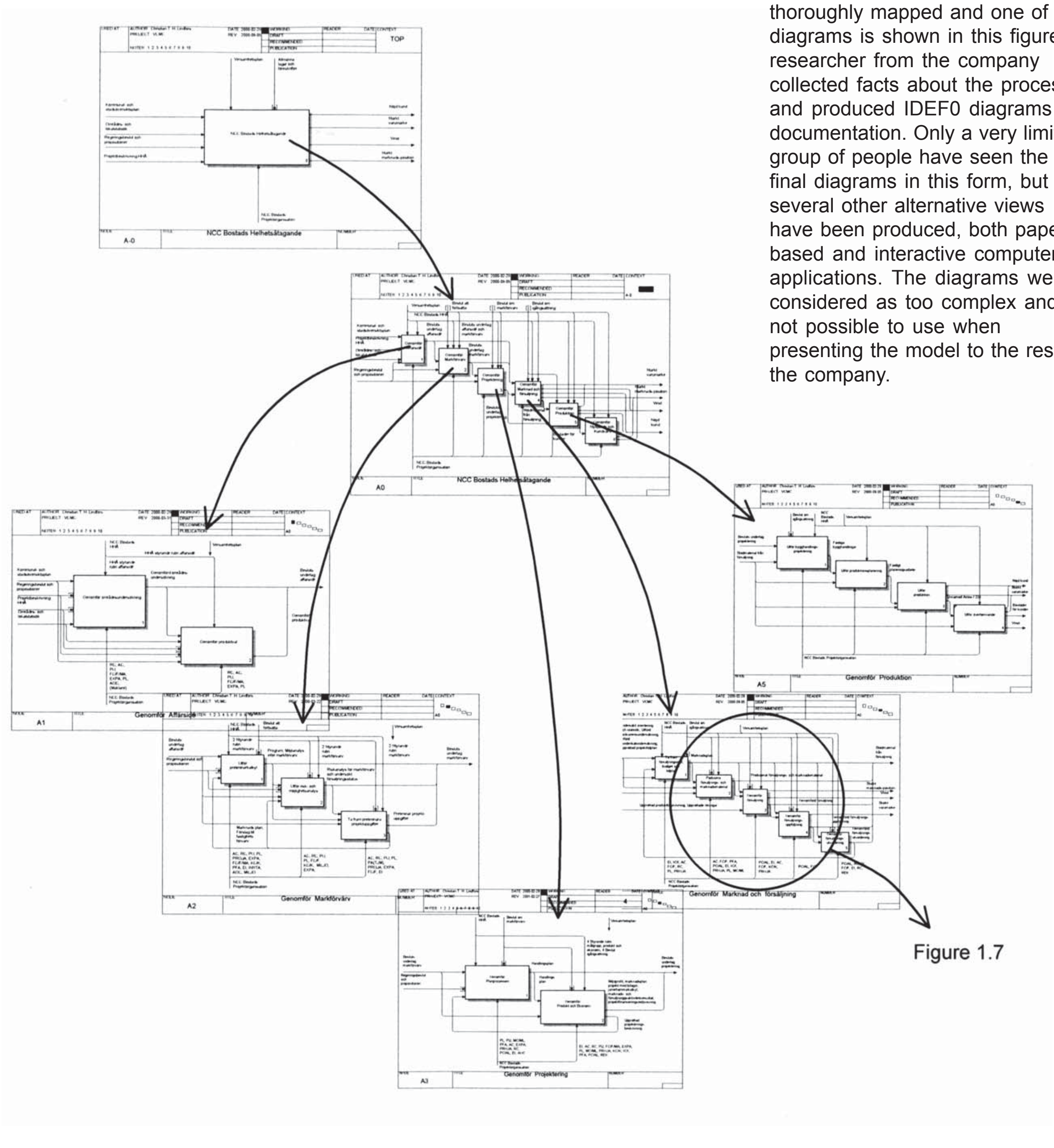


Figure 1.6 NCC is one of the largest construction and property-development companies in the Nordic region. The project development process has been thoroughly mapped and one of the diagrams is shown in this figure. A researcher from the company collected facts about the process and produced IDEF0 diagrams as documentation. Only a very limited group of people have seen the final diagrams in this form, but several other alternative views have been produced, both paper based and interactive computer applications. The diagrams were considered as too complex and not possible to use when presenting the model to the rest of the company.

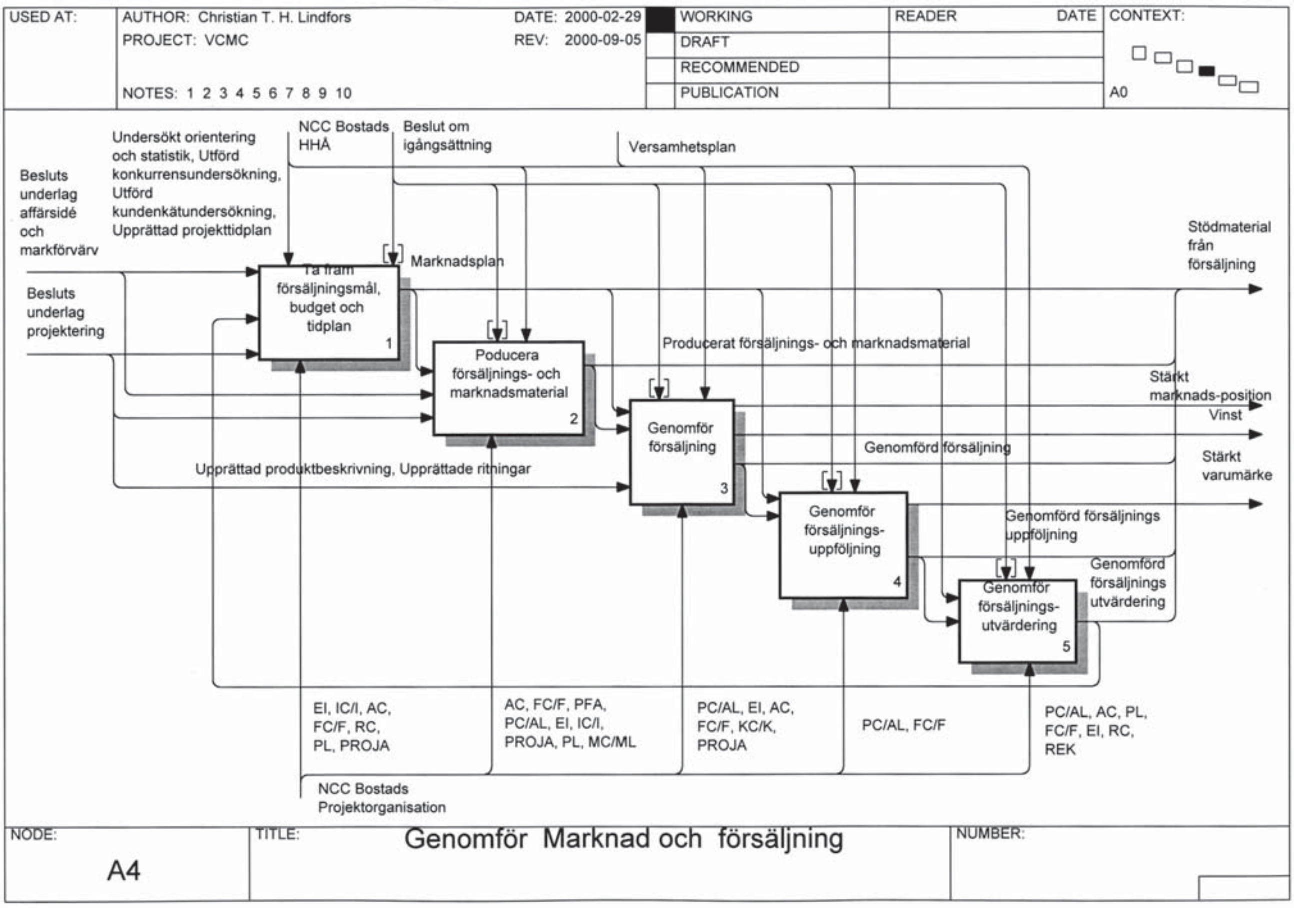


Figure 1.7 One of the diagrams in figure 1.6 displayed in a more readable size. The whole model consists of aproximately 30 diagrams of this type.

A rough classification of the examples in figures 1.3 to 1.6 is provided in table 1.1. All initiatives are multi-purpose projects and a consequence of this is that the different models have several representations.

Table 1.1 Purposes or arguments for modelling of business[3] processes (Eriksson and Penker 2000). The process models in figures 1.3 to 1.6 are presented in this table to indicate the types of initiative they represent. The classification is rough but it shows that there is often more than one motive. Purposes number 5 and 6 are not reflected in the four examples. This can indicate that the models are not sufficiently developed to satisfy these purposes (or that this was not of interest in the particular project).

		BAA	AP Fastigheter	NCC	Process Protocol
1	To better understand an existing business.	x	x	x	x
2	To act as a basis for creating information systems that support the business		x	x	
3	To act as a basis for improving current business structure and operation.	x	x	x	x
4	To show the structure of an innovated business.	x	x	x	
5	To experiment with a new business concept or to copy or study a concept used by a competitive company (benchmarking on the model level).				
6	To identify outsourcing opportunities.				

All the different arguments for creating business process models express a common need – to communicate with those involved in the modelling process. This interaction is handled in various ways, depending on several factors. If the process model is produced only to define requirements before modelling an information system, the process model is designed by information specialists and is never shown to end users. In other cases the process model is the outcome of the modelling activity. An example taken from efforts with quality management systems is the work of establishing a common process for inspection of documents. This latter process model will be used frequently by end users and must be useful for its special purpose. It is evident that the design is heavily affected by the intended usage; the design must make the model useable.

1.3 Usable process models

A lot of effort is often put into process modelling activities and to the production of process information. It is the opinion of the author that the presentation of the work is often of subordinate importance. Figure 1.6 and 1.7 showed a model that has not been used by ordinary practitioners. The creators of the model did not want to publish the model since they thought it would be useless. The modelling technique that was used in this case produces very complex models. Software that supports this modelling method are made for modelling and not for browsing. Other modelling tools (supporting other modelling methods) may have functionality that for example easy produces swim lanes or colour coding in order to improve readability. It is however easy to make an uncomplicated process look complicated by using these predefined widgets.[4]

The questions are, what widgets exist and when and how should they be used. Researchers within human-computer interaction (HCI) have tackled the same problem, but for general computer systems. Both findings and research methodology from HCI are of interest when considering widgets for process information visualisation.

Human factors (also called ergonomics and human-factors engineering) have a broader scope than just human-computer interaction.[5] Usability methods apply well to the design of complex systems (Nielsen 1993), which is the concern for human-factors engineers.

The models presented above in figures 1.3 to 1.6 are all heavily reduced in scale. It is a recurring problem that the models are so extensive that standard-sized sheets of paper are not large enough for the text to be readable. This problem is even more evident if the model is shown on a computer screen. Information visualisation is a discipline that among other things works with the problem of presenting a large amount of information within a very limited area (Herman, Melancon and Marshall 2000). Interactive environments provide opportunities that a two-dimensional paper representation cannot compete with. Expanding and collapsing menus of various geometric forms are widgets that make hierarchical information easier to explore. Process models are often hierarchical in their composition; a process can be broken down into sub-processes, which can be broken down into other sub-processes and so on. Using information visualisation widgets for visualisation of process information is logical. It is of interest to know what interaction a user must be able to perform in order to explore and understand a process model and use it as a tool for problem solving.

Choosing between widgets for process information visualisation and human-process model interaction must be based on usability concerns. Applying Usability to design of process models implies new ways of handling the result of process modelling efforts. The information produced is not the final outcome, to be handed over to an information designer who will make it look neat. The modelling does not end until the process model has a usable design. Design skills in general and knowledge about the user in particular are things that, in combination, contribute to making process models usable. However, there are other answers that remain to be discovered.

1.4 Objectives and purposes

This thesis presents the result of 2.5 years of research. The concrete objective is to present three specific results:

1. A number of new concepts, the definitions and structures of which make it possible to discuss existing representations of models and to develop new, more usable visualisations.

2. To shed light upon the significance of human factors when implementing model-based thinking and decision making.

3. Evaluation of methods for visualisations of principally different properties of process models and methods for navigation in process information environments.

Development of concepts implies naming phenomena that can be observed in a user interface for process information. The designations are based mainly on accepted expressions from HCI, Usability and Information Visualisation.

A question with a wider scope is illustrated when the importance of human factors is investigated. This is about the need to let Usability make a contribution to construction IT research.

Conclusions about usability of visualisation methods are based on two user studies. One of the studies is thoroughly presented in this thesis. The other study has already been presented in a previously published paper (Berg von Linde 2000).

The third objective uses the concept of the process information environment, which in this context, for example, includes process-oriented quality management systems or systems for project management information.

By considering questions about usability of process models' user interfaces, two effects can be obtained. The first one is, by bringing in knowledge about graphical user interfaces of process models and how they can be made usable, the awareness and importance of user perspective can be strengthened. This may contribute to more successful process-based initiatives in the construction industry.

The other effect is that the research can contribute to making Usability a more accepted concept within construction IT research and it can have an impact when it comes to implementation of any kind of result that is going to be converted into applications that will be used by the industry.

1.5 The research question

The research question that was the point of departure for this research project was initially formulated as:

> How shall user interfaces of process models be designed to be useful to practitioners of the construction industry?

Four terms in this sentence need to be discussed in order to make the research question unambiguous. The first term is "user interface". A user interface is the contact surface between user and system. The second term is "process model", which has been thoroughly discussed earlier in this chapter. The third term is "useful", which has also been introduced in this chapter and will be further discussed in chapter 3. For the moment it is sufficient to note that usability is about how well users can utilise the functionality of a system. The research question ends by mentioning the target group: practitioners of the construction industry. In this case practitioners are defined as persons who are not researchers or otherwise specialised in process modelling. The ordinary project manager, structural designer or architect without previous knowledge of processes and process modelling is the target group of the research question.

To decide if a process model is useable or not, distinguishing features of usable process models must be identified, as must methods for testing usability of process models. An exhaustive answer on the research question also needs to present a process that produces usable process models.

Hypotheses

Two hypotheses are used to find the answer to the research question. None of the hypotheses are traditional in the sense that they can be formulated as statements. The first hypothesis is a suggestion for a simple structure of concepts that describes visualisations of process information. The hypothesis is that these concepts are sufficient to describe a user interface and that it is also possible to use them to develop and evaluate user interfaces. The other hypothesis consists of a prototype combining a number of widgets that all aim at making the visualised process model usable. Evaluation of the hypotheses is done in a continuous chain. Use of the structure of concepts is exemplified in connection with its development in chapter 4 and is also the basis for the prototype. User studies are used to evaluate the prototype, which verifies that the implemented widgets actually contribute to increased usability.

The three objectives of the research presented above correspond to the hypotheses. Objective number one is connected to hypothesis number one, and objectives number two and three have a connection to the second hypothesis.

Problem owners and stakeholders

Defining the problem owners of the research project and its stakeholders will automatically cause demarcations but will also make it easier to evaluate the relevance of the research question. Owners of the problem are defined as those who work with process information and have a need to communicate this to others in a project, a company or a whole line of business. This definition is somewhat vague, but this is because people with varying backgrounds and positions in the construction industry create process information and process models. Stakeholders of the research are also a large and difficult to define group. The term “process owner” has been coined within the domain of process management, and persons who have this role can be regarded as stakeholders of this research. Process owners who cannot visualise processes have difficulty communicating with those who deliver or receive results of the processes as well as with those who perform the activities comprising the process.

The number of researchers and research projects within the area of process modelling for the construction industry is limited. Researchers active in the area are obvious stakeholders and receivers of knowledge developed in this research project.

Relevance of the research question

Research with connection to the construction industry about process modelling and process models has a relatively short history. Research about process modelling methodologies as such has not been done to any great extent by construction-related researchers. One example of an extensive development and research project that has used a process modelling approach is a project called Process Protocol. It is a British project which aims to develop a generic process model applicable to the British construction industry. Usability issues have not been discussed in the project judging from available documentation. A Swedish example is represented by the development project Förvaltningsinformation 2002 (Facilities management information 2002), which is being performed with support from the national research and development program IT Bygg och Fastighet 2002. Competent and experienced researchers and consultants have developed a solid body of information with receivers in facilities management companies. Presented process models are described with methods that proved to be useful in other industrial applications. The objective is that the research presented in this thesis will give tangible support when implementing projects like Process Protocol and Förvalningsinformation 2002.

A complete presentation of the objectives of this research has been done above. The relevance and validity of these objectives have been secured by the researcher's experience as a practitioner in the area as well as by research workshops and meetings with industrial representatives. This background work is not presented in detail in this thesis.

Generic value

Results and discussions presented in this thesis are primarily intended to be applicable to the construction industry and its research communities. However, it is possible to make generalisations across these borders. This is partly due to the fact that investigated dimensions in many cases are independent of the test persons' professions, professional skills and professional experiences. The independence also includes the content of the examined models. Both user group and application can be regarded as independent variables and thus admit generalisation across lines of business.

A subset of the result of this research will be knowledge about how usability of graphical representations can be tested. The objective is not to present generic methodologies for examination of usability, but since usability is a fairly unknown concept among affected research communities, it is necessary to explain basic methodological issues. This is for the most part done in chapter 2. Methodological discussions about usability have a generic value that goes beyond research about process modelling and models.

1.6 Structure of the thesis

The thesis consists of six chapters:

1. **Introduction** This chapter gives a brief background to the problem and presents objectives and purposes of the research.

2. **Method** Methodological issues concerning the research are presented in this chapter. Usability studies and prototyping are discussed in depth.

3. **State of the art** The research combines a number of different knowledge domains. These are introduced in such a way that aspects connected to the research can be understood. An attempt is made to show how the combination of the knowledge domains helps in solving the research question.

4. **Process visualisation** This chapter develops concepts that can be used to describe interactive environments for presentation of graphical process information. It is also shown how the concepts can be used to construct an information system for process information.

5. **Learning from reality** A number of the concepts developed in chapter 4 are implemented in a prototype. This prototype is tested in a usability study. The procedures of the study and the results are presented.

6. **Discussion and conclusions** The results of the research are discussed and critical questions are asked and answered. Finally, the main findings of the research are presented and some suggestions for further research are made.

Notes

1. Half dimensions are used in an intuitive way, representing hypertext or hyper-graphics. Dimensions that are not integers represent mathematically something very different; see for example fractal geometry.

2. This definition is meant to support both the engineering and the social science view of processes. Later in the thesis a functional modelling method (IDEF0) is chosen to exemplify process models in a prototype. The working definition is intended to embrace this particular way of describing processes as well. If the whole process model is considered as a system, input and output can be regarded as states of the system, even if the model cannot express time.

3. The word ”business” is not unambiguous. In this context a broad definition is adopted in which profit or predefined goals are not prerequisites for a business. A construction project or activities to capture customer requirements are considered as types of businesses which can be described with process models.

4. Components in a user interface are often referred to as widgets. For example, a drop-down menu is a widget. A process model is composed of various design elements. An activity can be represented by an arrow with a caption inside, activities can be grouped into swim lanes to indicate that they are performed by the same organisation, thin arrows can connect one activity with another one indicating output and input, etc. Such design elements are referred to in this text as widgets.

5. The term “human factors engineering” is used to designate equally a body of knowledge, a process and a profession. Woodson et al. (1992) define human factors engineering as “the practice of designing products so that the user can perform required use, operation, service, and supportive tasks with a minimum of stress and a maximum of efficiency”. “Human factors engineering” is the preferred term in the United States, whereas the prevalent term in Europe and most of the rest of the world is “ergonomics” (Woodson et al. 1992). The term “human factors” was used with a somewhat different meaning during the 1920s and 30s, and the term “ergonomics” was officially introduced in 1949 when the Ergonomic Research Society was founded (Ivergård 1982).

HCI and Usability are knowledge domains closely connected to human-factors engineering, but with a narrower concern. HCI and Usability are discussed in chapter 3 State of the art. In this thesis “human factors” is used as a generic term for knowledge about human performance, behaviour and training in man-machine systems, the design and development of man-machine systems and systems-related biological and medical research. See for example the paragraph about the objectives of the research on page 13.

2 Method

From a methodological point of view the most important contribution that this research makes is to exemplify methods for examination of usability. Examining proceedings from major conferences in construction IT shows that usability is very seldom discussed. Likewise, qualitative studies, and particularly user studies, are seldom performed. A tradition has been developed in relation to construction of models and prototyping. The following discussions about methods show how this research combines prototyping and user studies to investigate issues regarding usability.

Methodological questions concerning the usability study are given a relatively thorough discussion, among other things to inspire others to take to a similar approach in other contexts.

The chapter ends with a short discussion about the role of prototypes in research and its significance in this research project.

2.1 Usability study

User studies are performed to gain an understanding of what happens when a user is put into a certain situation. In this case, users are studied who, with help from information in a process model, try to solve predefined questions. Since it is primarily usability issues that are of interest, the user study can be designated as a usability study.

Evaluation of a user interface can be performed both with and without involvement of users. An evaluation without participation of users is called an expert review. Examples of different types of expert reviews are heuristic evaluation, guidelines review, consistency inspection, cognitive walkthrough and formal usability inspection. Common to all these methods is that they are performed by experts with knowledge of and experience with development of user interfaces. Within the particular area of user interfaces of process models exists no basic knowledge that is needed to be able to perform these methods. One unanswered question is, for example, what heuristics should be examined in a heuristic evaluation of a process model representation?[1]

To understand what affects the usability of graphical representations of process models, qualitative studies are needed. The type of qualitative study that was considered the most suitable in this case was user studies combined with observations and interviews. These studies need relatively large resources but can generate results that can be used to perform expert reviews in a later phase, which needs considerably less time and resources. However, no expert reviews have been performed in this research project.

Within the research domain of usability there is a tradition of working with qualitative methods and particularly user studies. The discussion below about the usability study that was performed in this research project is based on experiences from earlier user studies (Berg von Linde 2000) and methodological descriptions by, among others, Nielsen (1993, 1994) and Shneiderman (1998).

Objectives of the study

The objective of the user study is to gain understanding about how different principles for visualisation of process information contribute to making process models usable. The user study is a part of the testing of the hypotheses in this research. The prototype that is the subject of the study has been constructed on the basis of the concepts that have been developed to describe user interfaces of process models. The prototype and the concepts themselves represent hypotheses, and the user study is a method to test these two hypotheses.

Selection of test persons

It is possible to consider a number of dimensions when the test persons are to be selected. These dimensions are:

Novices – experts

Skilled computer users – unskilled computer users

Good domain knowledge – poor domain knowledge

Information producers – information consumers

Novices are users who have never before used the tested system, or who seldom use the system. An example from another context is users of ticket machines at a long-stay car park at an airport. Many users come across this user interface very seldom and can be regarded as novices. This must not prevent those who park their cars every week at this car park, experts, from using the ticket machine effectively. In this user study the focus will be on novices. Experts already have working user interfaces, even though there are many possibilities for development for them as well.

The dimension "skilled computer users – unskilled computer users" must be examined in both directions. The whole gamut of computer users must be represented in the user study. Examples of consequences for design of user interfaces are use of de facto standards for management of multiple windows, design (in terms of both colour and behaviour) of icons and design of objects that can be clicked on. Unskilled computer users have no previous knowledge to start from when they confront new applications and user interfaces. The advantage of de facto standards is small in this case.

The study aims to investigate how practitioners in the construction industry manage to use different kinds of process representations. The content of the test models belongs to the construction domain, but within this domain a more accurate subdivision can be made. For example, it is possible to perform a division starting from professional roles, see figure 2.1.

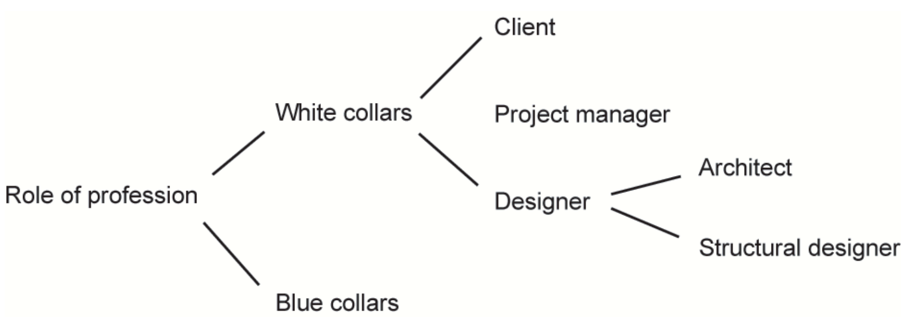


Figure 2.1 Division into domains on the basis of professional roles. The figure is neither complete nor consistent, but exemplifies the principle.

Only persons with domain knowledge participate in the user study. Therefore, the domain must be narrowly demarcated or the test model must have a content that pertains generally to several sub-domains. The user study applies a combination of these measures by choosing test persons who have white-collar positions and by using a model content that is generic enough to make it possible to consider all the selected test persons as having domain knowledge.

The dimension "information producers – information consumers" is studied only in one direction, from the consumer perspective. This is consistent with the decision to study only novice users, not experts, as expert user interfaces already exist. The same is true for user interfaces for information producers.

Number of test persons – reliability

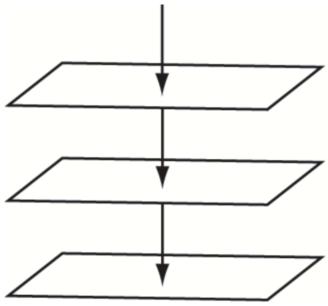
The number of test persons affects the reliability of the study. There is a trade-off between a good selection (a proper distribution between the dimensions of the study and a sufficient number of test persons) and the need for resources to perform the study. A 90 percent confidence interval and a standard deviation of 15 percent were obtained in a user study with 13 test persons, who were all expert users (Nielsen 1993). If the test persons are novice users a slightly higher number of test persons is needed to obtain the same accuracy. These experiences show that relatively few test persons are needed to achieve satisfactory reliability.

Reliability is, however, a general problem for user studies, for one thing because of great individual differences between the test persons. This is possibly a more severe problem when quantifiable differences are measured than it is when indications of problems are investigated. The study performed in this research project deals with the latter approach.

Validity

Validity is the question of whether the study really measures anything of relevance to usability of a real process model in a real situation. Understanding the methodological issues is actually the only way to get good validity (Kjaer Jensen 1995). Factors that can affect the validity in a negative way include the selection of inappropriate test persons, application of the wrong test tasks, omission of normal time pressure or other performance-reducing factors, etc. Confounding effects can also affect the validity of the study. The test model can, for example, be presented in such a way that the design of the test equipment causes behaviour that has no connection to the subject of the study, the process model

Figure 2.2 The user's path to understanding.



A fundamental problem with a study of this kind is that the effects that are registered cannot without a certain degree of reflection be connected to circumstances or properties of the examined system. Figure 2.2 shows the two levels that the user must go through to be able to assimilate the content of the model, level three.

Level one is the user interface of the computer application. This is the part of the interface that the user uses in order to manage the application as such. At this level, de facto standards[2] are used as much as possible.

Level two is the user interface of the process model. This level is composed partly of design elements given by the modelling method used, and partly of the widgets for interaction with the process model that are made available to the users. The third level is the actual content of the model.

In a test situation, deficiencies on levels one and two can be concealed by the fact that the previous knowledge of the test person is enough to solve the test tasks. The opposite effect can arise if the system that the model represents is unfamiliar or is regarded as complicated by the user. The user study performed in this research project aims only at investigating how level two affects the usability of the information system.

The procedure to isolate observations to level two is mainly:

To construct tasks that need understanding of properties that are managed on level two,

to perform the study in such a way that problems on level one are minimised (for example by providing necessary instructions about how to operate the user interface) and

to ask the test persons to check questions immediately after the test session to clarify why difficulties came up.

As well, level three can cause results that are difficult to interpret. The content of the model can differ from the test person's picture of reality, i.e., the mental model of the test person can differ from the model. The content of the model can even be regarded as incorrect, which can lead to the test person's considering himself unable to solve the test task.

Test model

Extent and content are the two most important aspects to consider when choosing a test model. The extent should be large enough so that the model contains enough objects and levels to make the model sufficiently complex. The content of the model should be familiar to the test persons, who are supposed to have domain knowledge. The basic model has, in this case, been constructed with IDEF0. This method was chosen since it generates most of the data that are of interest to visualise.

The business that was chosen as the subject of the process modelling was the project development process at NCC Boende, a major Swedish contractor. The business has been modelled by Christian Lindfors, a PhD student and process developer at NCC Boende. The described process is a business process and what is modelled is primarily information flow rather than physical things or logistical issues. The process ranges from design of business concept to delivery to customer. The abstract level of the model, its level of detail and the fact that it covers a complete business process makes it ideal as a test model.

Level of computer support

The author of this thesis has earlier performed user studies on entirely paper-based user interfaces. A two-dimensional paper has obvious limitations when it comes to offering interactive possibilities. Filtering, zooming and linking of relevant information, for example, are impossible to display in a paper-based user interface. To be able to evaluate most of the identified widgets it is necessary to use a computer-based user interface as the test environment. A paper-based user interface has the advantage of eliminating problems on level one (see figure 2.2). In connection with the discussion about validity, a strategy for dealing with problems on level one was presented which should outweigh the disadvantages of a computer-based user interface compared to a paper-based one.

Ethical considerations

Test persons are also human beings and of course cannot be exposed to unreasonable exercises. The manufacturing industry often uses destructive tests. For example, a chair can be loaded until it collapses. Even if it would be interesting for the research project it is hardly acceptable to stress test persons to a point of breakdown. During a user study a situation is inevitably created which can be experienced as uncomfortable. The test person often feels that he must make a good impression, and he is unwilling to reveal that he does not underst-and or that he has problems solving the tasks. Below is a list of things that a test administrator should consider when performing user studies (Nielsen 1993, 1994 and Shneiderman 1998):

Before the study

Emphasise that it is not the test person that is subject to the test but the test application (the prototype).

Explain that the application is not error free. It is a prototype that can display incorrect information and have functional deficiencies.

Make clear that the test person is fully entitled to interrupt the test session at any time without any motivations.

Explain how the test will be documented, which is particularly important in this case since sound recordings were made.

Make clear that all test results will be confidential.

During the study

Let the test person succeed with the first task.

Never express that the test person has failed in any way or is acting slowly.

Interrupt the test session if the situation feels too stressful or uncomfortable for the test person.

After the study

Finish by telling the test person that the test session was successful and made an important contribution to the research project.

Never play back recordings without approval from the test person.

In addition, it can be a useful exercise to have the test administrator take on the role of the test person in order to become more sensitive to how exposed and inadequate one can feel as a test person.

Data to be collected

Data about the test persons must be documented. Data of interest are name, primary professional role, number of years this has been the primary professional role, self-estimated ability to use computers, experiences related to process representations, and knowledge and experiences related to the process modelling method IDEF0. These data are collected at the test session during an open conversation. Previous knowledge about process modelling is in many cases hard to discuss since the concept of process is experienced as difficult to understand, even though the test person has relevant previous knowledge. The test administrator needs to be adaptable so that the test person does not feel awkward or uncomfortable before the test has even begun.

The test person is asked to think aloud while solving the tasks. Usability engineers usually call this method "thinking aloud protocol". Thinking aloud generally feels unnatural. One way of explaining what is of interest for the test administrator to hear is to let the test administrator demonstrate how it can sound when a person is thinking aloud, for example by launching a well-known Windows application and performing some simple tasks. Sound recordings are made during the whole test session to document the thoughts that are expressed aloud.

In addition to the sound recordings, the test administrator takes notes about interesting observations, both about what is said and, above all, how the test person is acting, such as what functions are being used and how they are used, what mistakes are made, what seems to be difficult to execute or solve, whether there are signs of nervousness or stress, etc. These notes are to be verified and clarified during the concluding discussion after the test tasks are performed.

After the last task (or as many tasks as the available time allows) has been performed, the test administrator initiates a discussion about what was experienced during the work with the test tasks. The discussion focuses especially on what tasks were experienced as difficult to solve and possible reasons for this, whether the meaning of the model was understood, whether process models representing a methodology are useful in practice and whether the test environment (the application) caused positive or negative feelings. It is during this discussion that the test administrator can verify the level to which the observations should be connected, see figure 2.2 and the discussion about validity above.

2.3 Prototyping

In research, the creation of prototypes can generally be motivated by different reasons. A prototype can, for example, demonstrate the practical possibilities for realising something that has been theoretically derived. In other cases the prototype provides a test bench that can be used in further studies. By thoroughly documenting the characteristics of the test bench researchers other than those who have initially constructed the prototype can carry on and make further developments starting from a highly technical level without losing speed.

Prototypes must, regardless of motives for their creation, be tested and verified. Verification can either be done by means of a theoretical discussion or by using empirical data, either qualitative or quantitative. This thesis presents a verification done as a combination of theoretical discussions and qualitative data.

The reason why a prototype was constructed in this research project was partly to show that the theoretical model developed in this research could be used to construct a properly working user interface. More important is the role of the prototype as a test bench. However, the objective is not to provide an underlying software construction that can be used for further development and studies. Issues concerning design of technical interfaces, module thinking, and the ability to change scale have not been considered at all. This is because the test bench is intended only for user interface studies. The test bench is intended to be an environment for simulation of realistic user situations.

Even though the prototype is not intended to offer a working underlying structure (dynamic connections to databases, generically designed and reusable widgets) but only the visible user interface, the development of the prototype takes considerable effort. This makes limitations on the functionality and the number of features inevitable. The number of features a prototype has is called the width of the prototype, and the degree of functionality of these features is called the depth of the prototype (Nielsen 1993), see figure 2.3.

A search feature, for example, can be implemented in such a way that a dialog window is available to allow users to type words to search for and send the words to the system, which, however, does not send any answer back to the user.

The prototype developed in this research project is described in detail in chapter 5 in connection with the presentation of the user study.

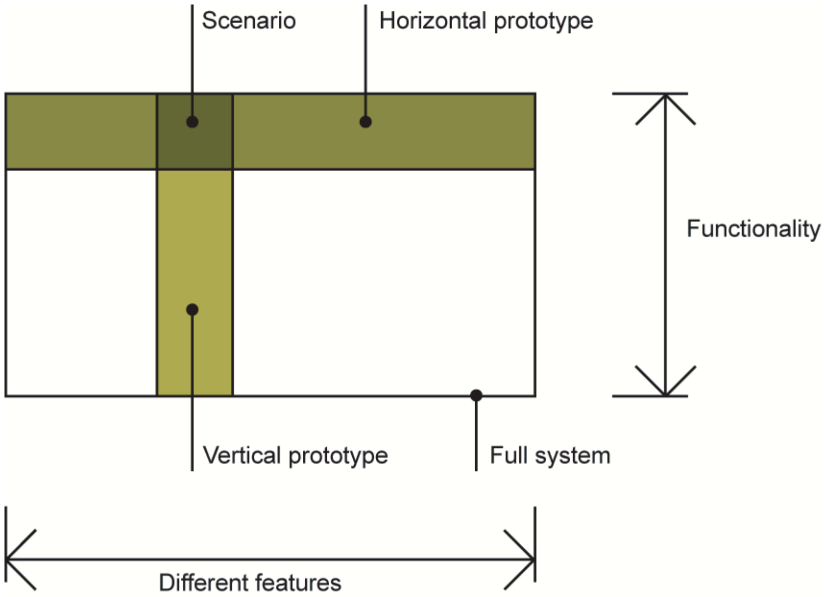


Figure 2.3 Strategies for construction of prototypes. Vertical prototypes have a few fully implemented features, whereas horizontal prototypes have a complete number of features with little functionality. A limitation is normally necessary in both directions by choosing a predefined scenario. Within this scenario is a satisfactory number of features sufficiently implemented to be possible to evaluate.

The prototype that is developed in this research is made to cover a relatively extensive scenario. Almost half of the features of a complete software system are implemented with in many cases full functionality. The area that depicts the scenario would be fairly large if it was drawn in this graphical model for prototype strategies.

The figure is revised from Nielsen (1993).

Notes

1. “Heuristic evaluations” implies that a small number of initiated reviewers examine a user interface to decide whether a predefined set of principles of usability, heuristics, is addressed (Nielsen and Mack 1994). A heuristic evaluation has the advantage of needing few resources to be performed. It can be performed in early stages, even on paper models of a planned system, and it often needs only a small number of test sessions to obtain a sufficiently accurate and useful result. There are several available compilations of heuristics; see table 2.1 for an example.

The concepts that are developed in chapter 4 in this thesis can constitute a basis of heuristics applicable to examination of the usability of process models. In addition, a complete list of heuristics for this purpose would also contain general heuristics of the type shown in table 2.1.

Table 2.1 Heuristics in no particular order, revised from Nielsen and Mack (1994).

Visibility of system status	Match between systems and the real world
User control and freedom	Consistency and standards
Error prevention	Recognition rather than recall
Flexibility and efficiency of use	Aesthetic and minimalist design
Help users recognise, diagnose and recover from errors	Help and documentation

2. Suitable de facto standards can be borrowed from Microsoft’s Windows environment and from the design tradition that has been developed so far on the web. Formal GUI guidelines are also available for Microsoft applications. The design of web pages has been developed by leaps depending on more flexible page description languages and methods (HTML, style sheets, etc.) in conjunction with more developed browsers. The development has not been managed solely by professionals and initiated designers but also by interested amateurs, who have been able to publish information on the Internet. It is thus hard to distinguish what are de facto standards and what are not, and what are the result of well-thought-out design strategies and what are just rush jobs. Nielsen has discussed and shown examples of good and bad web design in his book *Designing web usability* (Nielsen 2000). Significant for a book in this field is its perishable character. Widgets that earlier were difficult to implement technically can, after further development of the browsers’ functionality, become every web designer’s possession. Animations have long since been considered bad from a usability perspective, but because of generally increased bandwidth and an increased distribution of plug-ins like Flash and others, this is changing.

3 State of the art

This research is based on a number of different knowledge areas. The most important are information visualisation, usability, human-computer interaction, cartography and process drawing. To explain state of the art for this research project is to explain how these different knowledge areas can contribute to making process models usable. The knowledge area of process modelling may at a first glance be regarded as the most important. But that is not the case. Process management researchers have treated issues like usability and design sparingly if at all. The same is true for the existing documentation of process modelling methodologies. Discussing state of the art from a process modelling perspective alone is thus of little value. It is more interesting and exciting to discover the other disciplines.

3.1 Information visualisation

Some professionals are more accustomed than others to working with pictures as problem-solving tools. It is obvious that an architect uses images to carry out his work and to communicate with colleagues and clients. The importance of visualisations for other professionals is in some cases less evident. The medical doctor examines diagrams visualising the activity of a patient's heart and on that basis makes a decision about appropriate treatment. A diagram is an external representation (Ware 2000) and it extends and alters the cognitive process. What is difficult to imagine is sometimes possible to describe visually by means of paper and pencil. There is a well-known Chinese proverb which reads, "A picture is worth one thousand words". What is less well-known is that the proverb was made up by an advertising executive representing a baking soda company and the correct translation should be “A picture’s meaning can express ten thousand words” (Lester 2001). The fictitious Chinese proverb is depicted in figure 3.1.

Figure 3.1 The original “Chinese proverb” from the **1920s** baking soda advertisement (Lester 2001).

The misinterpretation is unfortunate, since it cuts out the two most important words of the proverb: *meaning* and *express*. Visualisations extend the memory and amplify our problem-solving capacity. Visualisations can be regarded as external aids, things that makes us smart, as Norman (1998) formulates it.

A working definition of visualisation can be found in Card et al. (1999):

> **Visualisation** the use of computer-supported, interactive, visual representations of data to amplify cognition.

From the same source, a working definition of information visualisation is formulated as follows:

> **Information visualisation** the use of computer-supported, interactive, visual representations of abstract data to amplify cognition.

Visualisation is concerned with non-abstract data sets. The data have an obvious spatial mapping because the data describe a physical and tangible reality. Examples are visualisation of weather phenomena or a structural design. Informational visualisation, on the other hand, is concerned with abstract data. There is a need to express data that do not have an obvious spatial mapping, i.e., economic figures and the structure of computer software. Process models clearly belong to this concept. The term “information visualisation” has an inherent ambiguity since it can be used both as a verb (how to make a visualisation) and a noun (the resulting artefact). The term is used frequently in this thesis and the context will in most cases reveal the meaning of the term; if not, the meaning is explicitly explained. Card, Mackinlay and Shneiderman (1999) describe visualisation as the mapping of data to visual form that supports human interaction in a workspace for visual sense making. This way of describing visualisation is illustrated in figure 3.2, a diagram that serves as a reference model.

The reference model shows how data can be transformed from raw data to visualisations. Transformations are made in several steps, starting with raw data which are mapped into data tables. Data tables are relational descriptions of data with attached metadata. Data tables are transformed into a visual structure by visual mapping. Visual

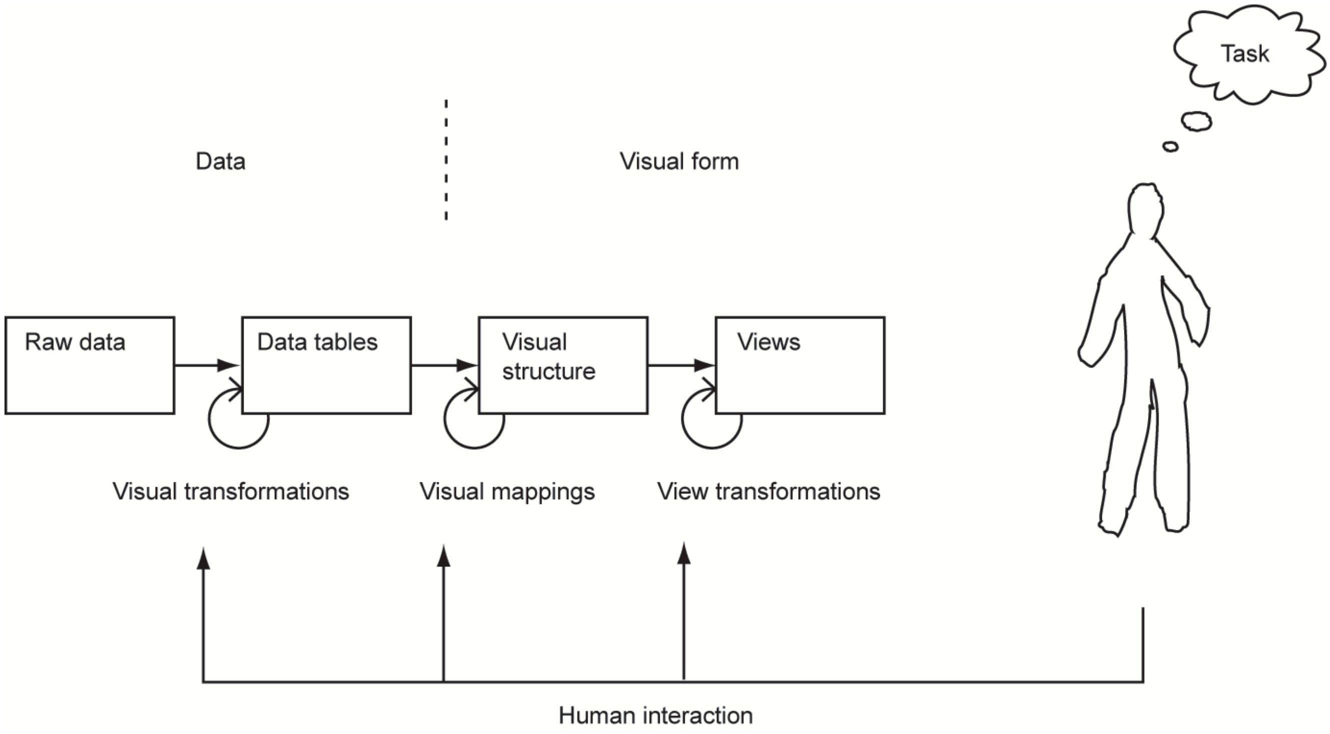


Figure 3.2 Reference model for visualisation. Revised from Card et al. (1999).

structures comprise spatial substrates, marks, and graphical properties. The last part of the reference model describes how the visual structure is transformed into views. Specifying graphical parameters, i.e., scaling and position, does this. All transformations of the reference model, that is data transformations, visual mappings, and view transformations, can be controlled by human interaction. For example, a user can change how the visual mapping is done in order to suppress details of less importance. Card et al. (1999) emphasise the importance of mapping data tables to visual structures. Data tables are constituted by mathematical relations and visual structures by graphical properties. Data with no obvious spatial mapping are analysed during this transformation from raw data to data tables. Of course, raw data can be visualised directly, but useful conclusions may not be drawn.

Applying this reference model to process modelling is straightforward. Mapping processes often start with identification of activities. This is usually done as a workshop session with participants working in the process that is subject to the mapping activity. The activities are written down in no particular order. The intention is to find out what activities exist, not how they relate to each other. The unstructured activities can be regarded as raw data in the reference model presented above. The next step is to identify the dependencies between the activities. One achievable method is to construct tables with all activities listed in the header, and in the cells below note how they relate to other activities. Relations can be expressed as “needs input from”, “is controlled by”, etc. Table 3.1 shows a simplified example of such an activity-mapping table. This table corresponds directly to the data table in the visualisation reference model.

Table 3.1. A simplified data-mapping table.

Activity	A1	A2	A3	A4
Relation	I{A2}, C{A3}	I{A3}	I{A4}	I{A2},C{A1, A2}

I = needs input from, C = is controlled by

The next step in the visualisation reference model, the visual mapping, can be performed in many ways. The challenge is to make all data from the data table visible in the visual structure and nothing else. Unwanted data can easily slip into the visual structure. Two dimensions are used to describe quality of mappings: expressiveness and effectiveness (Card et al. 1999). Expressiveness concerns the question of representing all and only the data in the data table, and effectiveness has to do with the creation of a mapping that can be perceived well by humans. The concept of effectiveness connects information visualisation to cognitive science and its application in human-computer interaction and usability. This is further discussed in connection with the unfolding of those knowledge domains.

The last transformation in the reference model dynamically creates views of the visual structure. Interaction with the visual structure makes it possible to extract more information from the visualisation than does examination of a static visualisation. Card et al. (1999) identify three common view transformations:

Location probes

Viewpoint controls

Distortions

Location probes use location to make additional information explicit. This can be achieved both by arranging objects in a meaningful way on the screen and by interactively displaying or highlighting information depending on what area the cursor is hovering over. Viewpoint controls transform the visual structure by using operations for zooming, panning, and clipping. The recurring problem with these operations is that the context is always difficult to maintain. Techniques have been developed to handle this problem; one example is called overview + detail. Distortions are view transformations that preserve context and display details at the same time. This approach is called focus + context views. Examples of focus + context views are the hyperbolic tree (Lamping, Rao and Pirolli 1995) and the table lens (Rao and Card 1994).

Human interaction is the final step in the reference model. Mapping between the stages can be controlled in different ways, depending on what kind of mapping the user wants to perform.

3.2 Usability

Nielsen (1993) has defined usability as a part of system acceptability. The concept of usability can be applied to almost any artefact as well as to abstract occurrences. A tangible thing like a door handle can be considered more usable than another one, and a web address may be more usable compared to another one.[1] Research on usability has, during the last 10 years, focused mainly on computer systems. Norman has shown in his *The design of everyday things* (Norman 1998) that usability is an issue for anyone who designs something that will be used by humans. A system that satisfies all needs and requirements of the users is simply an acceptable system. Part of the requirements is about degree of usefulness. Usefulness is a matter of whether the system is capable of achieving the desired goal. Finally, usefulness is a matter of utility and usability, where utility covers the issue of sufficient functionality and usability relates to how well users can utilise the functionality.

The attributes of usability are considered by Nielsen (1993) as:

Learnability

Efficiency

Memorability

Errors

Satisfaction

See figure 3.3 for a graphical presentation of these concepts and their relations to each other.

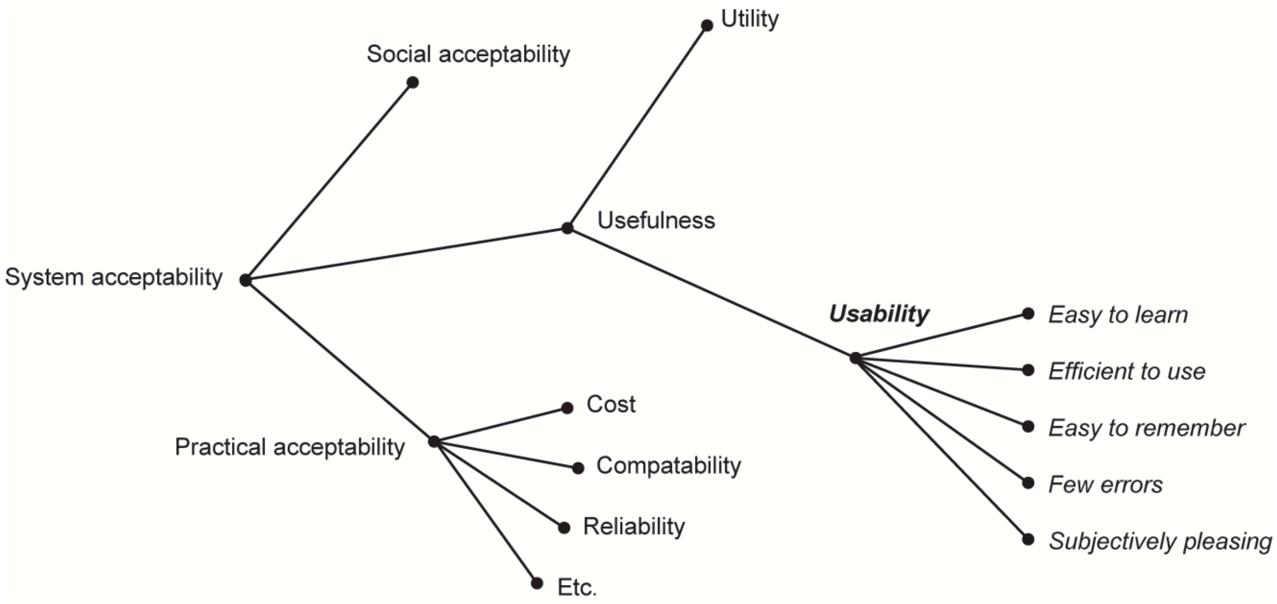


Figure 3.3 Usability and its attributes and relation to system acceptability, revised from Nielsen (1993).

Usability experts strive to measure those attributes. The quantitative school is of frequent occurrence and usability tests are often performed as laboratory tests while seconds, actions, and comments are quantified and countered. There are many usability evaluation techniques that have been developed to satisfy different demands on accuracy, time- and cost-effectiveness, etc.

Usability is a relative measurement. It depends on what task is to be performed and by whom. A system naturally performs better pursuing the tasks for which it is developed, and a user can only perform on the basis of her knowledge and skills.

This makes it important to formulate goals for a system that is subject to a usability test. Who are the users and what are the typical tasks that they want to perform?

Evaluation of usability can be done with or without user involvement. If users are not involved, an expert review is done. There are several types of expert review techniques, some of them listed by Shneiderman (1998):

- Heuristic evaluation
- Guideline review
- Consistency inspection
- Cognitive walkthrough
- Formal usability inspection.

The main disadvantage with expert reviews is that the expert may have extensive knowledge about usability theory but may not have an adequate understanding of the task domain or user communities (Shneiderman 1998).

Usability testing is an evaluation technique that, in contrast to expert reviews, needs user involvement. A usability test can either be done in a laboratory environment or at the place where the system is typically used. A cellular phone might, for example, be tested in a noisy street. The test administrator starts by setting up test goals and making a test plan. Test persons are selected according to how representative they are of the intended users of the system. Reliability of the test is to some extent dependent on the number of test persons used. A number of 10 to 20 test persons could in this case be enough, see discussion on page 24. Test tasks must be compiled. The test tasks should be as representative as possible of the intended usage of the system, and they should also cover the most important functionality of the system. During the actual test, an experimenter is needed to manage the test sessions.

Documentation and performance measurement can be done in several ways. Common methods are video and sound recording, logging of actions, and taking notes manually on paper. The thinking aloud protocol is commonly used to capture the users' intentions regarding performed actions. It also reveals major misconceptions that the system may cause. Usability testing generates lots of data, both quantitative and qualitative.

Analysis of the data is performed in a way that suits the nature of the data and the test goals. For example, if a test goal is to find out if search time has been reduced in a new system, quantitative data that describes search time are statistically analysed and, if meaningful, presented graphically.

3.3 Human-computer interaction

Despite a fairly long history (relevant references go back as far as 1960), an agreed-upon definition of the area of human-computer interaction (HCI) does not exist. A special interest group on computer-human interaction within ACM (Association for Computing Machinery) has proposed the following working definition (ACM SIGCHI 2001):

> Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

Usability and human-computer interaction are both overlapping and complementary disciplines. Human-computer interaction has its focus in interaction between one or more humans and one or more computational machines. Computational machines are often embedded, e.g., videocassette recorders, cars, and mobile telephones. HCI is concerned with computational machines in the form of ordinary workstations and embedded systems. Designs for interfaces to devices of this type have many things in common and can be treated together. However, less interactive and more mechanical devices, such as a tin opener for example, are not part of HCI. The tin opener could, however, be considered an issue for the usability discipline, which is more closely connected to general human factors.

In conformity with usability, human-computer interaction is an interdisciplinary area. Some of the disciplines are computer science, psychology (particularly cognitive psychology), sociology and anthropology, and industrial design (Dix et al., 1998). The ACM definition of HCI opens up for even more disciplines by stating that "major phenomena surrounding" computer systems are part of the human-computer interaction discipline. Using HCI to develop user interfaces to graphical process models is, in other words, in accordance with the tradition. Studying HCI is a shortcut to relevant knowledge within all the disciplines that HCI embraces. Cognitive psychology, for example, is an extensive field of science with many well-developed theories. HCI researchers have managed to extract theories that are relevant and applicable to the problem of interaction between computers and humans. There has also been a transfer in the other direction; cognitive psychologists have studied the learning of systems, the mental representations by humans, and human performance on such systems.

3.4 Cartography

There are two main scientific approaches to cartography; the communication-oriented and the representational (MacEachren 1995). Several researchers have described cartography as a communication process graphically (Board 1967, Kolácný 1969, and MacEachren 1995). The structure of those descriptions is basically the same. A cartographer interprets the real-world geographic environment and makes a map that a recipient reads; the recipient develops an understanding of it by relating it to pre-existing knowledge (figure 3.4).

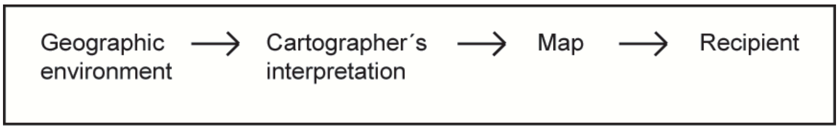


Figure 3.4 A depiction of cartography as a communication process. After MacEachren (1995).

According to MacEachren (1995), the communication paradigm regarded cartography as a process of communicating spatial information. The process had inputs, transmission, and reception of information and could thus be analysed as a system. This system had several filters that the information had to pass through on its way from reality to the map user. Reducing the filtering effects or the loss of information in the system could improve the map communication. This paradigm has been criticised and few cartographers accept it in its literal sense. One objective, which is also relevant when process maps are considered, is that the communication paradigm does not support all possible ways a map can be used, it just supports the conveying of information. Of much more interest is the map as a tool for problem solving and information exploration and development.

The communication-oriented approach to cartography used behaviouristic methods and measured narrowly demarcated attributes. MacEachren (1995) calls it “map engineering” and it has similarities with usability engineering, with Nielsen as its most influential advocate.[2] As a reaction to this somewhat mechanical view on cartography, a more holistic school developed which allowed scope for artistic values. In their extreme form, scientific approaches to cartography are considered impractical or irrelevant. Cartography is an art rather than a science, and the content of a map is more important than the information it contains.

A third view is represented by the combination of art and science. No scientific or non-scientific approach exists with respect to how maps work. MacEachren (1995) suggests that cartography is about representation, which is a statement that may seem to be obvious, but he also claims that an understanding of how representations work will help us to understand maps.

3.5 Process drawing

Traces of demand for readability can for example be found in textbooks' descriptions of how processes can be graphically illustrated. The recommendations for the design are based more on a need to create uniform diagrams than on promoting readability and usability. This section discusses two different ways of describing processes. Both methods are thoroughly documented. The design recommendations of these two methods are examined in this section.

Business process innovation, business process reengineering (BPR) and process management in general are management concepts that have needed methods for describing processes explicitly. The methods that have been discussed in this context have, in many cases, been relatively simple and scantily documented. Davenport, who has been important for the development of the management concept business process innovation, has summarised demands on tools for presentation of processes. Finally, this section discusses briefly some other descriptions of methods for graphical visualisation of process models.

Integration definition for function modelling, IDEF0

In the standard of the IDEF0 method (NIST 1993) there is a section that describes how IDEF0 diagrams should be graphically designed. There are several instructions solely given to increasing the readability of the diagrams.

The instructions illustrated in figure 3.5 appear to be based on common sense. Knowledge about cognition and visualisations can nevertheless be used to show that there are generic principles which, used to design IDEF0 diagrams, will lead up to these instructions. The so-called Gestalt laws, for example, are a basis for explaining why instructions like a and c in figure 3.5 are of importance.[3]

a) Draw arrows along horizontal and vertical lines, not diagonally or as curves (except at corners).

b) If an arrow is long, label it twice.

c) Connect open-ended boundary arrows to show all the places affected.

d) Space parallel arrows adequately.

e) Lay arrows so as to minimize crossing.

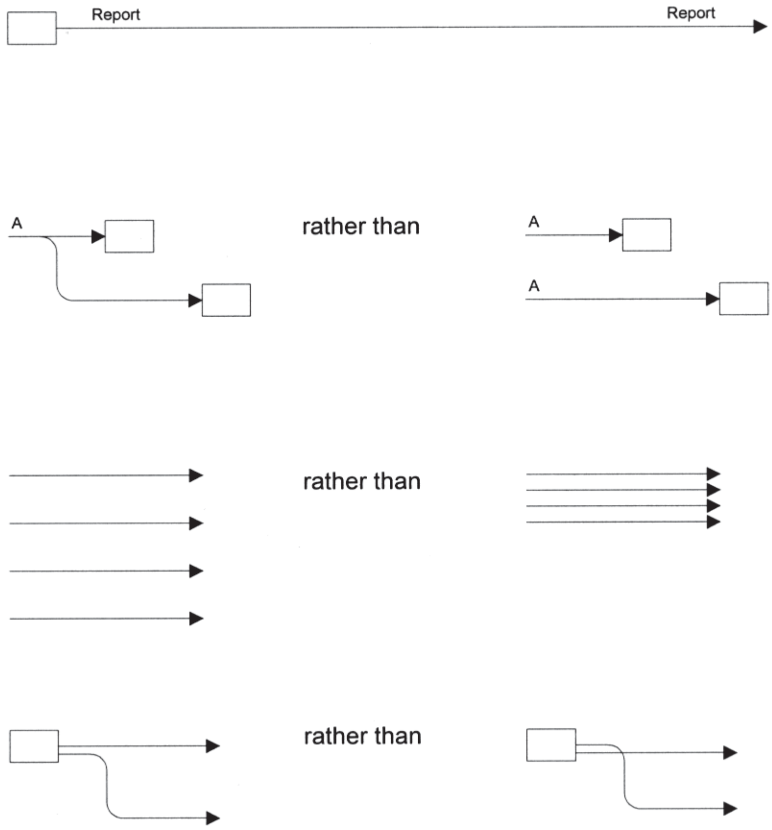


Figure 3.5 Examples of design guidelines revised from the IDEF0 standard (NIST 1993).

Feldman (1998) describes a quality measurement rule that he calls Fog Factor Testing of IDEF0 diagrams. Design aspects of the diagrams are analysed in order to evaluate readability of the diagrams.

The modeller answers a number of questions to obtain input to calculation of the fog factor. Each fog factor question had a numerical answer, which is summed to a total fog factor. Each fog factor has an individual maximum value to prevent a diagram from becoming unreasonably complex. The questions and maximum threshold values are presented in table 3.2.

Table 3.2 Definition of the fog factor. Revised from Feldman (1998).

Factor	**Maximum threshold**
F1. Number of boxes on the diagram	6
F2. Number of input arrows entering each box	3
F3. Number of control arrows entering each box	4
F4. Number of output arrows leaving each box	3
F5. Number of arrow forks or joins	No maximum
F6. Number of arrow crossings	No maximum

Fog factor = F1+F2+F3+F4+F5+F6 < 50

The modeller answers a number of questions to obtain input to calculation of the fog factor. Each fog factor question had a numerical answer, which is summed to a total fog factor. Each fog factor has an individual maximum value to prevent a diagram from becoming unreasonably complex. The questions and maximum threshold values are presented in table 3.2.

The total fog factor is simply the sum of the six sub-factors. The diagram is too complex if this value exceeds the suggested maximum value, and it needs to be revised. The idea is that the maximum value can vary from one project to another depending on users and application area for the process model.

This method measures, in a relatively mechanical way, the readability of a model. Feldman is of the opinion that the best way of testing the communicative ability of a model is to observe the reactions that the users of the model express (Feldman 1998).

The IDEF0 standard does not regulate in detail the design of the diagrams. Instructions are formulated in a general way, which makes it possible to design the diagrams quite freely but still conform to the standard. The result of this has been that different software for process modelling with IDEF0 can produce diagrams that vary considerably. For example, many modelling programs allow the use of colours, which is not described in the standard.

Another detail not described by the standard is the bending radius of the arrows. When an arrow changes direction it must be done as a 90-degree bend. No sharp corners are accepted, rather, smooth bends are used. How this bend should be designed can be regarded as a subtle detail without any importance for the readability of the diagram. Since IDEF0 models are usually are drawn by means of a modelling program it is the software that has determined how the bending radius should be performed. The bending radius has nevertheless been given, almost without exception, nearly the same size by most of the modelling programs. The explanation for this is probably that there exists an optimal bending which is desirable with respect to the diagram space, readability and aesthetics. Figure 3.6 shows a cut-out from an IDEF0 diagram and how different bending radii affect readability.

Process modelling with UML

Eriksson and Penker have made an extension to UML (Unified Modelling Language) for modelling of business processes. This extension has since then been accepted as a part of the UML standard. This extension describes how processes can be modelled in order to make easy connections to ordinary information modelling with UML. The description of the process extension includes no discussion at all about how the graphical process models should be designed to facilitate readability.

Figure 3.7 illustrates a generic process diagram drawn according to Eriksson and Penker's extension to the UML standard. A process diagram drawn according to this method is usually made up of several process symbols like the one shown in figure 3.7. More information can be added to the diagrams by arranging the processes in swim lanes or making so-called assembly line diagrams. The UML standard

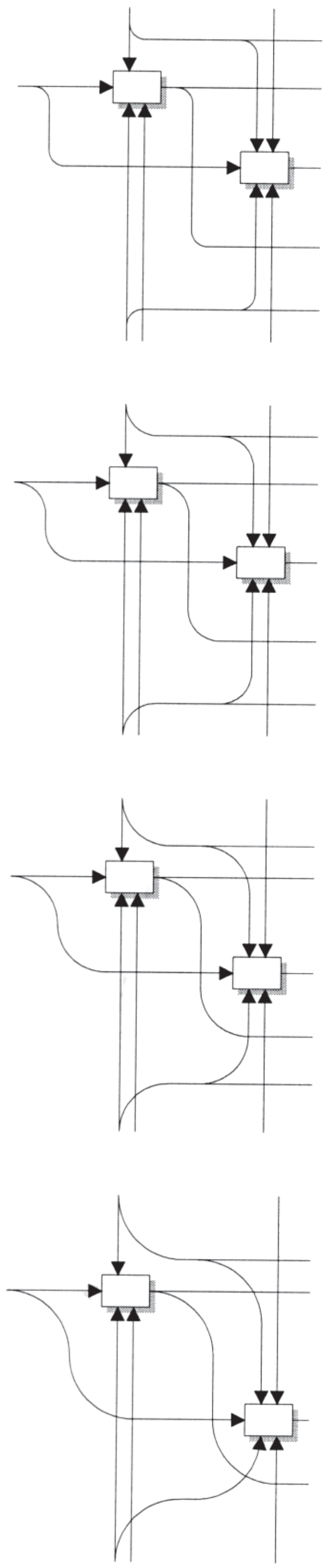


Figure 3.6 Playing with the bending radius. The diagram at the top is closest to the bending radius that has somehow become a standard for IDEF0 modelling software.

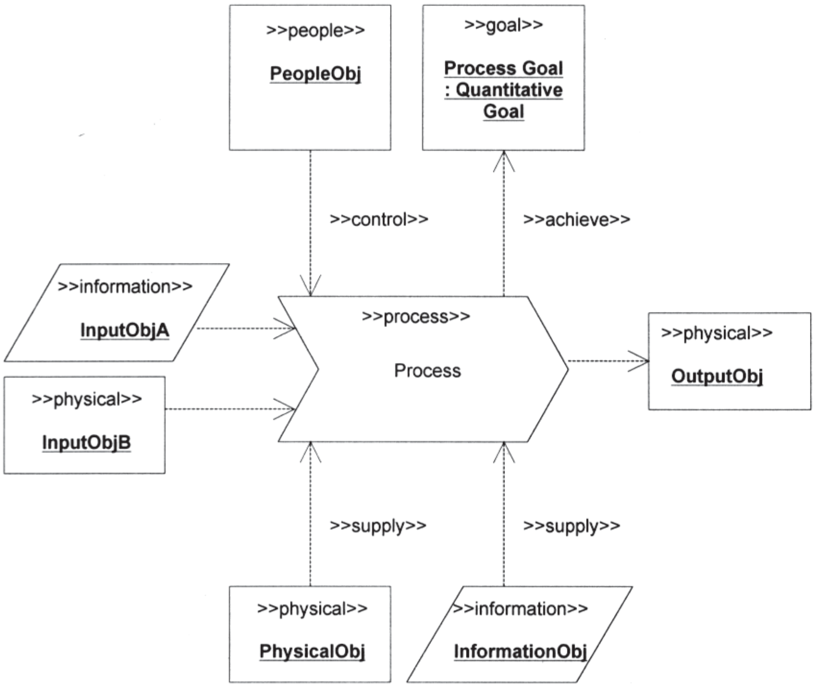


Figure 3.7 A generic UML process diagram drawn according to the process extension by Eriksson and Penker (2000).

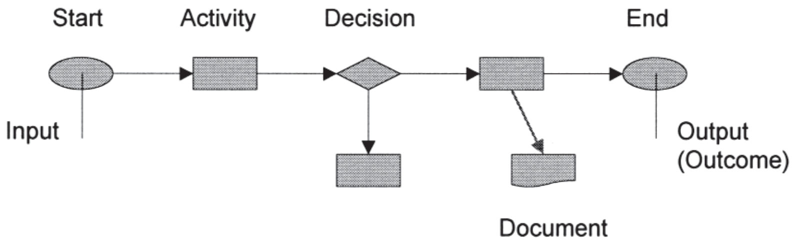
also contains other types of diagrams that can be used to describe processes, i.e., statechart diagrams that illustrate states of objects, and sequence diagrams that present interactions between objects.

It is evident that UML is developed by software engineers and for software engineers. The process extension made by Eriksson and Penker is not different in that sense. The processes are given a look that appeals to engineers and resembles drawings of technical devices. The diagrams are probably not suitable to present to anyone other than information engineers. UML doubtlessly achieved widespread acceptance among software engineers, and the process extension will probably be no exception.

Other design guidelines for process models

Both IDEF0 and UML are well-documented methods. The area of process management has in many cases used more loosely described methods. The graphical notation is less strict, but the diagrams are shaped according to the case in point. There are mainly two types of diagrams that appear in different forms, namely work flow diagrams and matrix flow diagrams (Rentzhog 1998), see figure 3.8.

Figure 3.8 Two basic types of diagrams commonly used in process management initiatives. At the top a flow diagram is depicted and below a flow matrix diagram. There are standardised ways of using and drawing the symbols, among others an ANSI standard



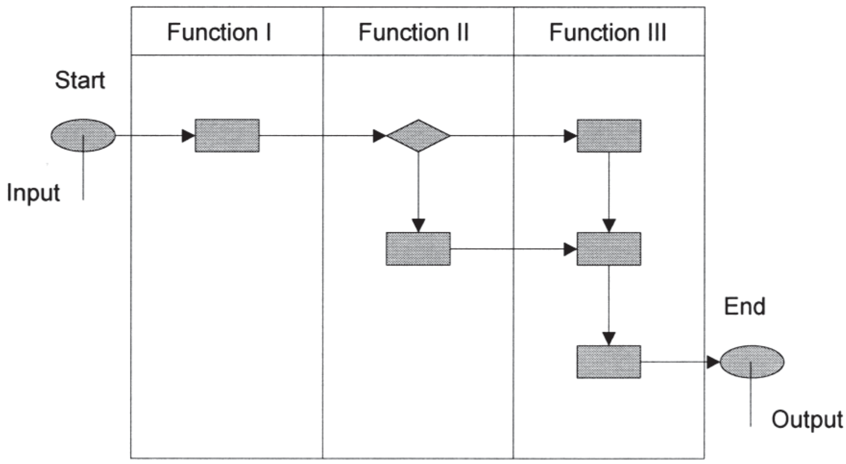


Table 3.3 Some of the requirements on a process design tool identified by Davenport (1993). The first four requirements are connected to usability and readability issues. Item number seven is notable, since it almost describes the concept of decomposition, which is further developed in chapter 4.

1. Graphically portraying the process steps.
2. Presenting a highly interactive and preferably graphical user interface.
3. Running live simulations and producing real-time graphical output.
4. Fast and easy to use at a high level.
5. Depicting the flow of materials and information between each step.
6. Accepting and portraying flow rate, resource and time consumption, and capacity and/or trigger information for each process step.
7. Rolling or exploding the steps of the process in a hierarchical fashion.
8. Identifying key bottlenecks and constraints in the process.
9. Linking to data and procedure modelling aspects of the CASE tool set to be used in IT-based system design.

It is mandatory to mention bar-charts in this context. Bar-charts or Gantt schemas are probably the planning technique that is most used by the industry and particularly by the construction industry. Little has been written about usability aspects specifically connected to bar-charts. Bar-charts are considered by traditional textbooks about construction management as a superior planning tool because of their simplicity and communicative abilities (Harris and McCaffer 1995). Others are more sceptical and are of the opinion that the technique belongs to the sequential world, which is no longer desirable (Wenell 2000)

Davenport's book *Process innovation: Reengineering work through information technology* (1993) has influenced the development of the BPR concept. The importance of working with graphical pictures in order to facilitate communication is emphasised by Davenport. However, no detailed descriptions of how to realise this in detail are given. Davenport says somewhat vaguely that "any consistent set of easily understood symbols will suffice". It is notable is that Davenport identifies demands on systems for modelling and presentation of processes and calls attention to user aspects and graphical design as important demands, see table 3.3. A true usability perspective is not presented, but the objective is, if anything, to create graphics that are attractive enough to appeal to senior management. In this context it is said that no tool exists that can satisfy these demands, but systems were soon to be developed.[4]

To sum up this brief look at process drawing, the topics of usability and more developed reasoning about design of graphical process models have not been covered extensively by literature on process modelling or process management.

3.6 Combing the knowledge domains

Information visualisation, usability, human-computer interaction and cartography are disciplines which in some cases work with identical methods and theories but in other cases are far from overlapping. A mixture of these disciplines forms a rich palette that is ideal for tackling the problem of making process models usable. Figure 3.9 shows an integrated picture of the disciplines and what their contributions are to this work.

Figure 3.9 shows three types of objects: users, applications, and models. The application contains two types of graphical user interfaces (GUI), one for the application itself and one for the process model.

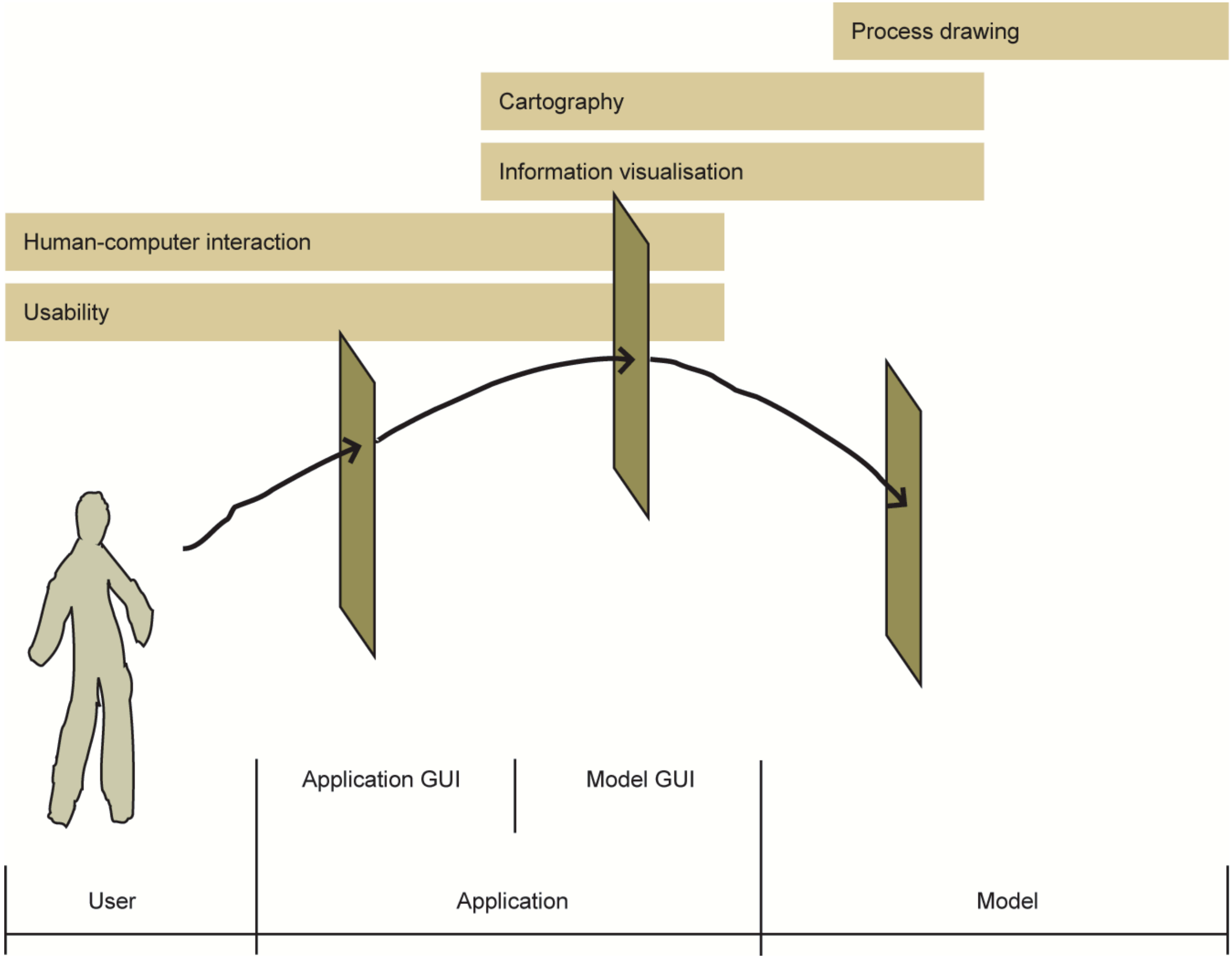


Figure 3.9 Knowledge domains that contribute to the understanding of the human-process model interface.

This thesis is primarily concerned with the application section in the middle of the picture.

The word "application" is used in figure 3.9 and should in this context be interpreted as an artefact that makes it possible to explore a model. It can either be computerised or paper based. A process model can be defined in several ways, i.e., by graphical symbols, tabular data, and narrative text. The mapping from model to application is dependent on the nature of the process description as well as on what application is required. The process of mapping structured or unstructured information into diagrams or graphical user interfaces is described by disciplines as graphic drawing, information visualisation and cartography.

Developing the computer application is a separate problem. There are several examples of dedicated development processes with focus on usability, thoroughly described in order to be useful in commercial software projects.[5] Such processes are beyond the scope of this research and do not help solving the research question.

A graphical user interface needs a computer application that stores the data, produces the images on the screen and the responses to interaction from the user. Algorithms for graphical drawing comprise a discipline of their own and will not be examined in this thesis.

The graphical user interface can be designed with help from guidelines developed by HCI or usability researchers and practitioners. The guidelines can be presented as a pattern language[6] or a list of heuristics.

To determine the degree to which the user interfaces (both computer and paper based) are useful in view of the users' prior knowledge and the tasks that need to be done, evaluation techniques are needed. Evaluation techniques are central in both human-computer interaction and usability. Paper-based interfaces, however, must rely on usability evaluation techniques, since HCI methods are mainly focused on computer based systems.

Cognitive psychology is embedded into several of the disciplines marked out in figure 3.9. In this research, issues connected to cognitive psychology are considered in both design and evaluation of user interfaces.

Notes

1. Long after the World Wide Web was considered to belong to everyone, characters such as the tilde (~) were common in web addresses. Today, fortunately, we have learned to live without such characters, which few of us could readily produce with the keyboard.

2. See for example Nielsen (1993, 1994).

3. The Gestalt School of Psychology was founded in 1912 to investigate the way we perceive form. "Gestalt" is the German word for pattern. Ware (2000) presents seven Gestalt laws, which were produced by the Gestalt psychologist. The laws are interesting since they describe in a simple and clear way many basic perceptual phenomena. The Gestalt principle of continuity, for example, states that we are more likely to construct visual entities out of visual elements that are smooth and continuous rather than ones that contain abrupt changes in direction. It is for this reason that all arrows in IDEF0 have curved corners at joins, forks and bends.

4. The author of this thesis is of the opinion that there still exists no software that lives up to these demands. It is eight years since Davenport made his list of demands and they are still appropriate requirements.

5. See for example The GUIDE-process described by Redmond-Pyle and Moore (1995).

6. An interesting example of pattern languages is Tidwell's *Common ground: A pattern language for human-computer interface,* see http://www.mit.edu/~jtidwell/interaction_patterns.html.

4 Process visualisation

An explorer's most important task is to convey experiences and discoveries from the journey to those back home. To explore worlds without sharing them with other people has no value for anyone but the privileged adventurer. It sometimes happens that doors open to completely new worlds. In those worlds phenomena exist that do not have any resemblance to the world we knew up to that moment. We realise that it is hard to describe something that no one else has seen or experienced before. It is simply because things and courses of events in the newly discovered world haven't yet been named. But it is not enough just to give phenomena names. Robust concepts arranged in a logical structure are needed to develop a useful understanding of the world. When the explorer has reported his observations, others must be able to follow on and draw conclusions about how the new world works. The discoveries may be so interesting that more expeditions should be made.

Combining knowledge about usability, information visualisation and process modelling opens a door that leads to a new world. Each of these knowledge areas has developed a serviceable world of concepts. The combined knowledge area, which is created in this case to develop usable process models, has no defined world of concepts. A conceivable way to establish this world of concepts is to start from the existing concepts and combine them while taking into consideration the demands made by the new application of the knowledge areas.

In this chapter, after the new world of the concept has been developed, it will be used to describe a number of examples of methods for visualisation and interaction with processes and their representations. Both well-known components of a user interface and totally new ones will be examined by means of the developed world of concepts. In conclusion it will be shown how the concepts can be used as guidelines when developing a completely new process information system.

4.1 Shelves for the mind

The world of concepts that is developed in this chapter will be applicable to representations of processes expressed on paper and as computer environments. Some of the concepts, however, will not be applicable to paper models, which have a considerably limited ability to offer interactivity.[1] Two different types of concepts will be defined. One type of concept concerns the objects that make up a model of a process. Examples of such objects are Input and Resource. The other type of concept describes the interaction with the model. These concepts are the tools that the user can use to utilise the model. The first type of concept is by way of introduction called Objects, and the other one is called Tasks.

A generic rule for information visualisation can be formulated as "overview first, zoom in and filter, then details on demand" (Shneiderman 1998). This is known by authors, film directors and architects. The procedures for realising this rule vary considerably among those professions. Nevertheless, the challenge is the same, which is to make a story purposeful and comprehensible. As a start, an overview is given. The context is explained by showing how relations to the surroundings are constituted as well as where the user is positioned at the point of departure. When the context is understandable we are ready to look more closely into our areas of interest. We move closer to some chosen area (zoom in) and, to make properties clearer and more perceivable, we switch off the properties that are not relevant (filter). When the picture appears clearly enough, we are ready to dive into the depths, to bring out the details about our area of interest (details on demand).

An example of a well-known application to this generic rule is the ordinary morning paper. The most important news items are presented on the front page of the paper with large headlines and short and easy to grasp texts. The reader is given an overview of what the newspaper will tell on this particular day. References are given to more detailed reading further on in the paper (zoom in), which is often divided into separate sections covering different subject fields, for example economy and culture. If the economy section is opened the reader can expect to find only economy-related news (filter). There is a lot of space inside the sections of the paper to elaborate in detail the news items that were only touched upon on the front page (details on demand).

Another example is the presentation of a virtual reality model. Presentations of VR-models are almost without exception started with a so-called over-flight. With a bird's-eye view, the model is approached and, in what looks like an accident, the flight ends in front of the entrance of the building. The doors open smoothly and let the user of the VR-model into the building.

Included in the generic rule "overview first, zoom in and filter, then details on demand" are some of the most important concepts for description of interaction between user and process model. These concepts belong to the types of concepts called Tasks. Shneiderman (1998) and Card et al. (1999) consider information visualisation in general and suggest that the list of tasks should contain (in addition to the concepts included in the generic rule):

Relate

History

Extract

Browse

Search

Read fact

Read comparison

Read pattern

Manipulate

Create

The last two concepts regard systems for developing and changing visualisations, which will not be a subject of this thesis. These two concepts will not be discussed further.

Relate is a task that shows the relationship or connection between different objects. An overall view can be shown in a user interface at the same time as more detailed views. The relationship between the detailed views and the overall picture, for example, can be visualised by means of colour coding. Displayed details are coloured with certain defined and differing colours. The explorer from the Windows environment uses a window divided into two vertical frames. The left frame shows the hierarchical structure of the file system, i.e., folders have sub-folders, which in turn have other sub-folders and so on. Files contained in the folder are displayed in the right frame. The folder that is open is depicted as an open folder (see figure 4.1) to show the relationship between the directory structure and the displayed files.

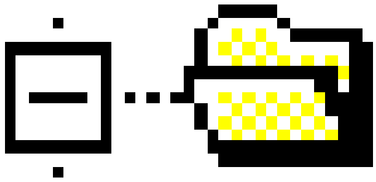


Figure 4.1 A detail from the well-known user interface of the Windows browser. The folder is open and the minus to the left is the option that closes the folder.

History allows the user to step back in his own electronic footprints. In complicated environments, it is often easier to go back to a familiar position and from that point try to find the way on. If there is no possibility to undo an action the user will experience an apparent stress and will probably not spontaneously try to explore unfamiliar parts of the environment (Nielsen 2000).

Extract can at first glance seem to be very much the same as a filter, but the two concepts represent tasks of a different order and level. By using filtering, properties in the whole system are suppressed. To extract is to remove or isolate a defined part of the considered system. The isolated system possesses the same properties when it is extracted as when it was an integrated part of its mother system. An extraction must not lead to simplification in any matter, except that the proportion of the system is reduced. All the other tasks can be performed on the extracted system. Filtering, for example, can be performed on an extracted system.

Browse is about enabling the user to explore the information system with relative ease in an unstructured fashion. Hypertext is a convenient way of creating easy browsing. However, it is not easy to create links that make a hypertext system usable (Lynch and Horton 1999).

Search can be done in many different ways depending on which methods for interaction are used. Words or concepts that the user wants to find can be input in different ways, and the result of a search can be presented in different ways depending on what is searched for and how the search result is going to be used. There is a division between methods that allow direct manipulation and those that do not. For example, a user interface that allows direct manipulation is one in which the user can clutch a piece of information with the cursor and drag it across the display and make peripheral parts come into the centre of the picture, which makes the information appear more clearly (see the hyperbolic browser later on in this chapter). An ordinary form in which the user can type a search word and send the question to the system for processing is an example of a user interface without direct manipulation. Some kind of spatial representation is needed to make direct manipulation possible. Otherwise the user does not have anything to interact with. For example, it is difficult to implement direct manipulation on one-dimensional text.

Read facts is related to how details are assimilated in the process model. Some of the objects that comprise the model are labelled with clarifying texts. These labels can either be continuously visible or can be visible/invisible or can change size depending on how the user chooses to interact with the model.

If two different states are displayed in different views it can in some cases be difficult to distinguish the differences between the states. Special kinds of tools for comparisons can exaggerate differences by marking objects that differ, for example, with certain colours. Degree of deviation can be displayed by using a defined scale of colours. Another way of facilitating comparison is to place transparent objects in layers above each other, which will effectively reveal any differences.

It is often interesting to trace patterns in a piece of information. Extensive models need tools that make it easier to find patterns. There are patterns of different kinds, for example repetitions, sequences and connections. To develop tools for tracing patterns calls for deep knowledge about the visualised piece of information. Generic tools are difficult to develop since the character of the pattern is highly important in deciding how it can be found and made visible.

Expressive needs

By way of introduction, it was mentioned that there are two main types of concepts, namely Objects and Tasks. The second concept, objects, can be considered as the expressive needs of the user interface. Concrete and visible objects are only a part of the user interface's expressive needs. Remaining needs are represented by the properties that the process model has, and those that it has to convey in order to be usable. In many cases process models have a hierarchical composition. Simple models have reasons to be flat, i.e., the whole model can be depicted on a single level. More extensive models need to be divided into several levels to be manageable. A process can, on its most comprehensive level, be represented as a small number of partial processes, which in turn can be broken down into other processes and so on. The user interface must make this procedure, decomposition, possible to visualise.

Processes are changes of state, courses of events (see the introductory chapter). The simplest way of depicting a process is to describe the states and connect them to each other in a sequence. More advanced process models have as well a need to express succession or sequence. There are, however, examples of process modelling methods that express sequence in an implicit way. Restrictions and demands in the model can reduce the number of possible sequences to only a few and thus implicitly describe the sequence of the process. This is applied, for example, by the IDEF0 methodology.

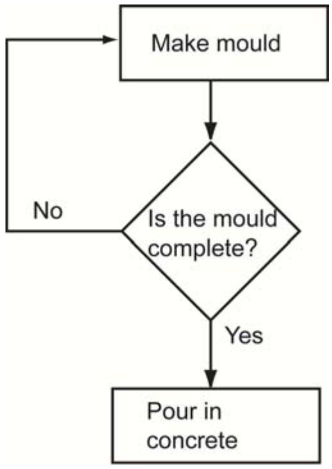


Figure 4.2 The decision point acts as a constraint. If the mould is not finished, the concrete cannot be put into the mould.

Processes interact with each other. For example, one process can deliver input to another process. Processes can also control other processes by setting up constraints that must be respected. In a process that describes manufacturing of a foundation of a house it must be clear that the mould must be finished before the concrete is added. Constraints can be expressed in several ways, for example as decision or evaluation points, often graphically illustrated as rhombuses; "if mould is finished then pour in concrete otherwise go to make mould" (see figure 4.2).

User interfaces of process models need to offer an overview. An extensive process model is practically useless if the comprehensive picture cannot be conveyed. In some cases this problem is addressed by drawing the whole process on one single level, with the consequence that the model takes up a great deal of space (see for example figure 1.3). Another way to solve the same problem is to display the whole process as a miniature model. No details will be distinguishable, but the main parts of the model will appear and give an overall picture of the model.

In conclusion, the user interface must also be able to house the concrete content of the model. A set of objects that represent different kinds of information does this. A sub-process, for example, can be depicted as a box, input and output as arrows etc.

This main group of concepts that by way of introduction was called objects are thus a collection of expressive needs. The expressive needs that are identified above are the following:

- Decomposition
- Sequence
- Constraints
- Overview
- A set of objects

If a comparison is made between this list and the previous list of Tasks we will notice a certain degree of overlap. Sequence and Constraints can both be placed on a level with the Task Relate. Overview is both a Task and an expressive need. A way of avoiding overlapping concepts is to make another classification. A group called Actions can be created, and to this group belong all Tasks and the Expressive need called Decomposition. "A set of objects" is then the only remaining item in the list of Expressive needs. The classification will be made up of two main groups. One group collects all objects and the other group the actions that can be done on the objects, see table 4.1.

Table 4.1 Concepts that describe an interactive environment for process models.

Actions		**Objects**
Overview Zoom Filter Details-on-demand Decompose Relate	History Extract Browse Search Compare Find patterns	Process Sub-process Input Etc.

In the remaining part of this chapter the presented concepts and their structure will be used to classify components of a user interface. The concepts can also be used to develop a complete user interface for presentation of process information, and this will be demonstrated in conclusion.

4.2 Methods for process visualisation

Inventiveness has been significant when it comes to developing widgets to user interfaces of ordinary computer applications. As process models are converted from paper format to different types of computer environments, more and more of those widgets have been brought to the world of process modelling. In many cases it is just a matter of making process models available on the corporate intranet. The first stumbling steps are taken by publishing screen prints directly from a modelling or drawing program. In this way two-dimensional models are published. The next step is to facilitate navigation. Hyperlinks are added to the graphics and a 2.5D model is created. Many reach this stage. It is at this point that problems appear. Some users complain about difficulties finding the details they are looking for, others that they do not understand the connection between different processes, and the administrator of the content struggles with the problem of spreading the actual content of the model.

A process information system published on an intranet as a number of hyperlinked graphical pictures manages at best to satisfy the Actions of Decompose and Browse. Eventually, it becomes possible to produce more detailed information about the process as linked documents, i.e., the Action Details-on-demand, are implemented. The user interface is undeveloped and has few of the tools that can make the underlying process model usable.

Below is an exposition of components of a user interface, widgets, that make it possible to perform one or several of the identified Actions.

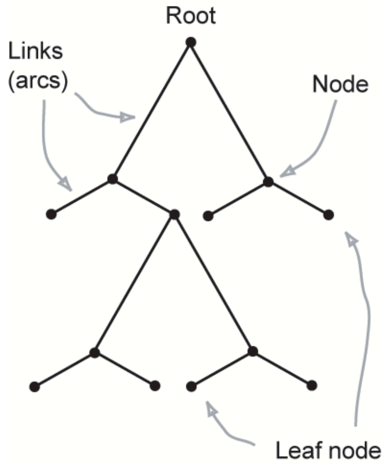


Figure 4.3 Components of a (vertical) node-link diagram.

Trees

Hierarchical data can be visualised as two types of tree diagrams. Those types are called node-link diagrams and enclosure diagrams[2] (Card et al. 1999). Node-link diagrams have nodes connected by links and can be vertical, horizontal or circular, see figures 4.3 and 4.4. A vertical diagram has its depth on an ordinal Y-axis, and the X-axis is nominal and is mainly used to keep the nodes separate. The horizontal diagram uses the axes in the opposite way. Circular diagrams use the R-axis to indicate depth, and the angle between the links is used to separate the nodes. One of the problems with node-link diagrams is that they require a great deal of space to show proportionately small amounts of data. If text that presents the nodes is going to be readable the space between the nodes must be ample. Because of this, the diagrams contain a lot of empty space. A way of making the diagrams more space efficient is to make use of interaction. A tree can be partly presented at first and the user can unfold the part that is interesting. Figure 4.5 shows an example from a web page that uses a horizontal tree for navigation.

Letting the user hide and show parts of the tree uses the space covered by the tree more effectively. The disadvantage is that the overall picture can be lost and the user needs some time to explore the whole diagram.

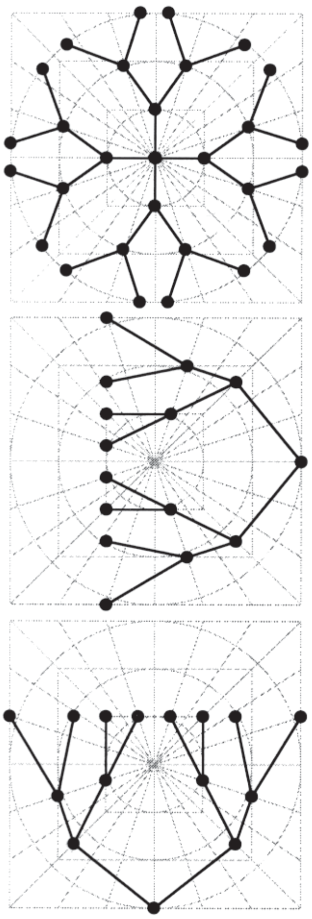


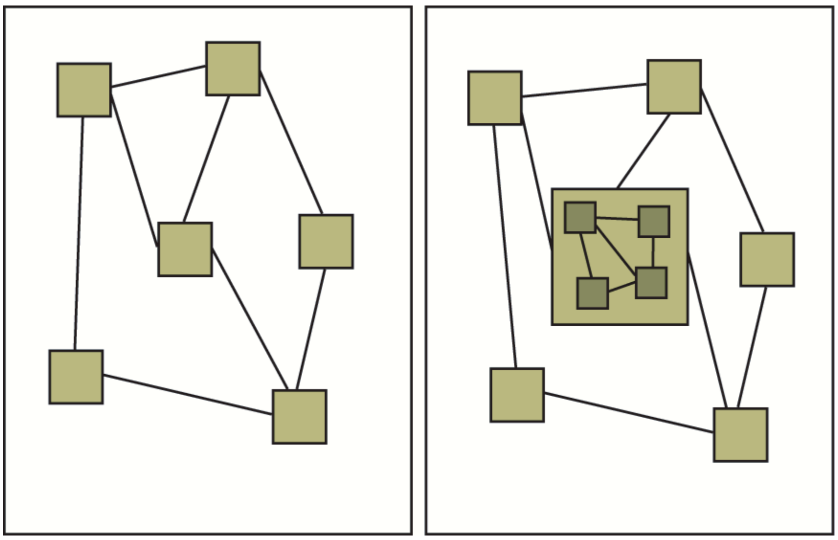
Figure 4.4 Vertical, horizontal and circular node-link diagrams.

Another technique, developed both to use the space effectively and to give an overview, is to make use of distorted views. The part of the diagram that is observed is depicted clearly, whereas other parts of the diagram are partly or completely faded out. The distortion can be linear in one or several directions or according to some mathematical

Figure 4.5 A horizontal tree provided on a web page as a navigation device. A limitation with this technique is that only one branch can be visible at a time. A click on another branch makes the first one collapse. The example is taken from www.constructit.se.

function which can give the impression of perspective effects. This way of using distortion to show details and context in the same view is called focus+context (Shneiderman 1998). Focus+context is based on the fish-eye technique, see figure 4.6. A creative development of the fish-eye technique has been made by Lamping, Rao and Pirolli (1995). By using hyperbolic functions the nodes can be spread out on what looks like a spherical surface. The hyperbolic browser shows initially the root node in the centre of the figure and depicts the rest of the nodes on the hyperbolic plane. The user can click on peripheral nodes,

Figure 4.6 The fish-eye view used on a network diagram.



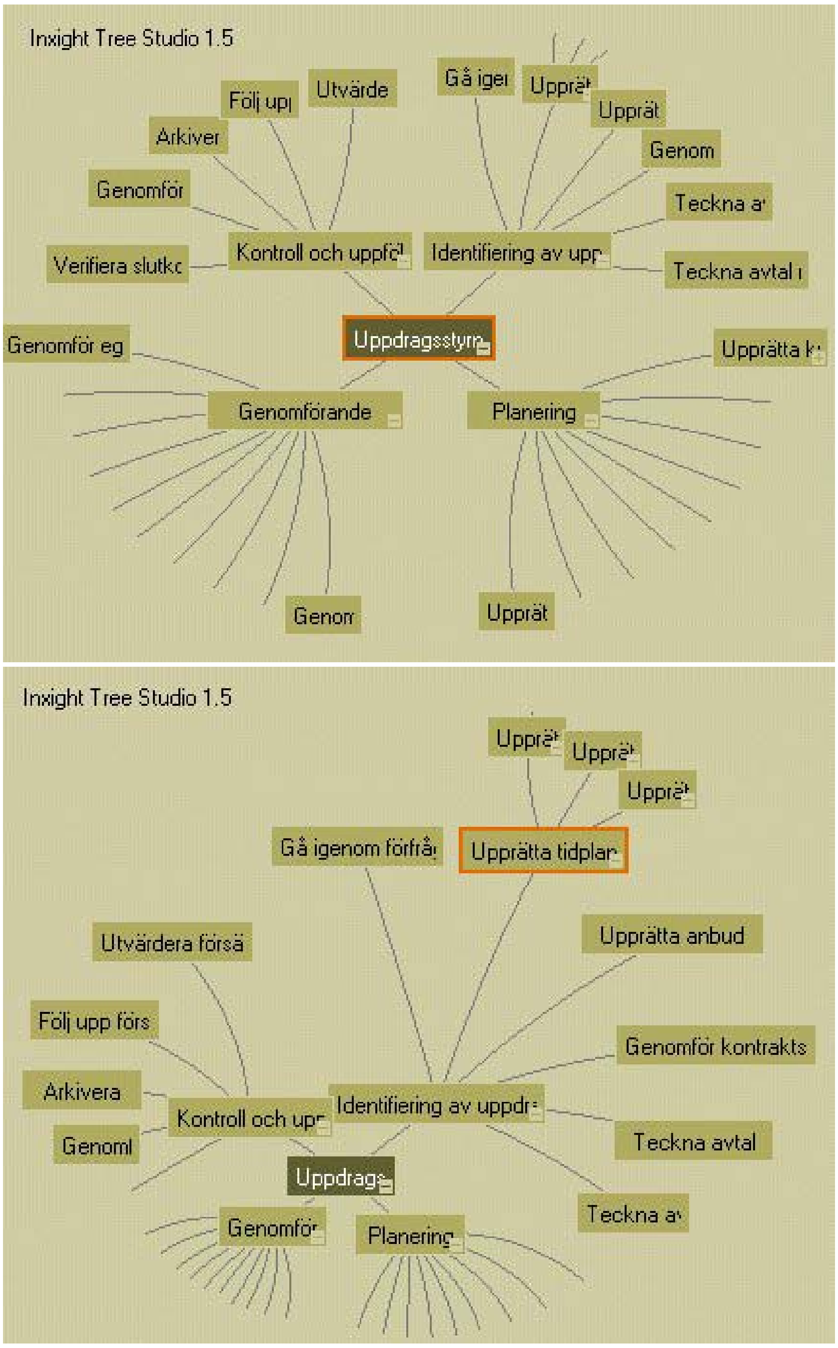


Figure 4.7 The hyperbolic browser implemented with Inxight Tree Studio. The tree shows the structure of a limited part of a quality system used by a Swedish construction consultant firm. The left image depicts the root in the middle of the diagram, and in the right image the focus has changed to a node in the upper right corner of the diagram.

which then move to the centre of the figure. The movement is performed in such a way that the user can follow how the node moves over the plane, i.e., the movement is animated and the nodes move like cities on a rotating globe (see figure 4.7).

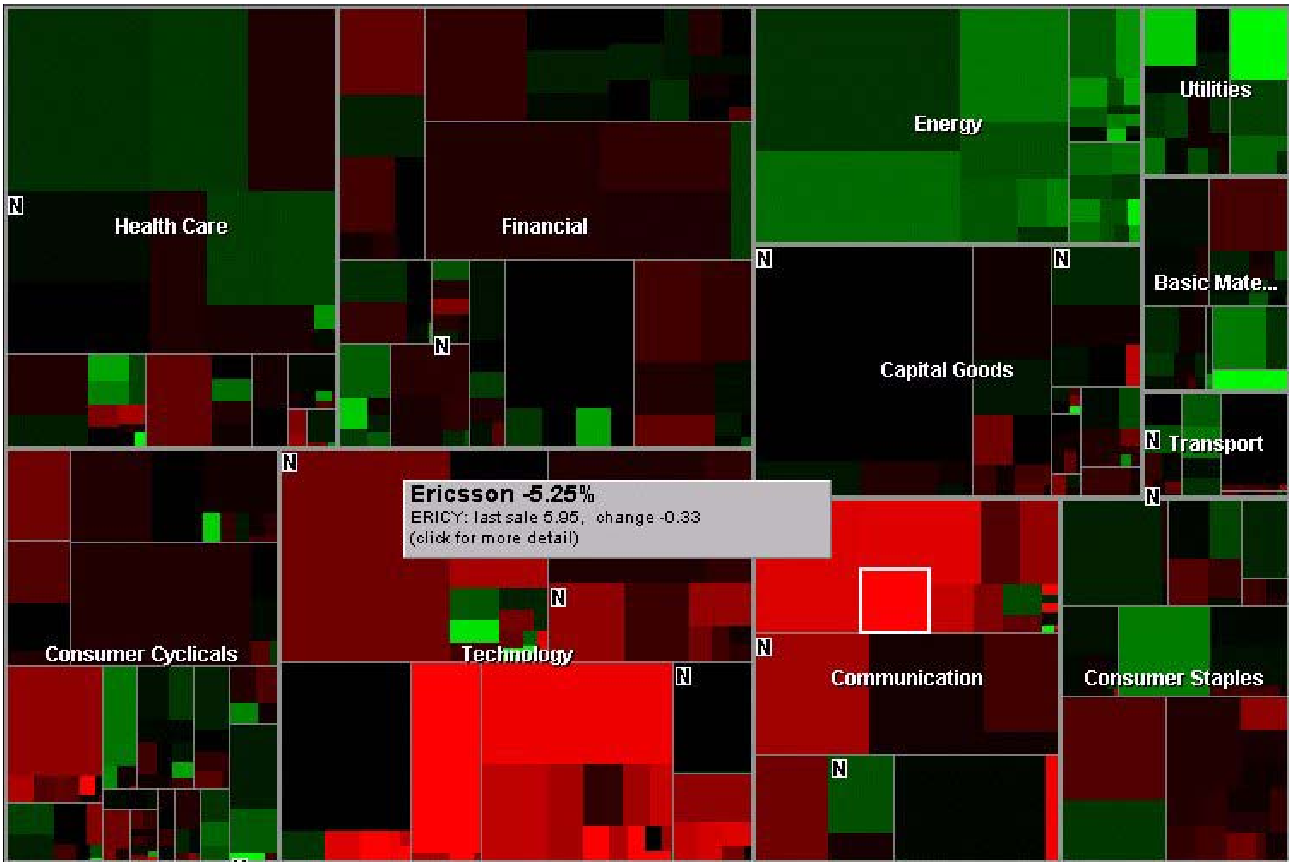


Figure 4.8 An example of a Treemap. The picture shows share rates at the New York Stock Exchange. The size of the squares is proportional to the market value. 11 main groups can be seen on this picture and Technology represents the largest value. Colours indicate changes, either since yesterday's closing or since any other day one might wish to compare with. Green stands for a positive change, red for a negative. The Ericsson share is selected and it can be noted that the current rate is 5.25% lower than yesterday's. This diagram can be found at www.smartmoney.com.

Node-link diagrams can also be drawn as three-dimensional diagrams. The advantage is that available space can be more effectively used if three dimensions are used instead of two. Examples of diagrams of this type are Cone tree and Cam tree diagrams. The links between the nodes are arranged in cone-like formations. Some of the nodes will conceal other nodes, making the diagram more space efficient. The cones can be rotated by the user so that all nodes can be examined. Studies have shown that users make fewer mistakes using 3D views to find information in tree structures compared to traditional flat node-link diagrams (Ware 2000).

It was mentioned earlier that there are two fundamentally different ways of visualising hierarchical data, either as node-link diagrams (connection) or as enclosure diagrams. Unlike node-link diagrams, enclosure diagrams fill the whole space. Johnson and Shneiderman (1991) have developed a type of diagram that they call Treemap. Boxes represent the nodes and the area of the boxes is a measure of the size of the nodes. This makes Treemap an effective way of visualising hierarchical, quantitative data. Treemap has, among other things, been used to visualise computer directories, libraries and economic data. Figure 4.8 shows an example that can be found on the Internet. To use the Treemap to greatest advantage, the information must not only be hierarchically organised but must also contain numerical data. A process model does not always contain quantitative information, and if Treemap is used the areas will be meaningless and cause confusion. If key parameters are measured in the process there is an obvious advantage to using the Treemap technique. In that case, the performance of the process can be understood in direct connection to the visualised process model. Another construction-related Treemap application is cost control in construction projects. Even relatively limited projects produce cost reports that are hard to take in. With a Treemap one need only take a short glance at the diagram to find out where the budget has been exceeded.

Based on these examples of how hierarchically structured data can be visualised and implemented in computer applications we can draw the conclusion that the field of application for tree diagrams is wide. The examples also show that the usefulness of the tree diagrams is very much improved if carefully chosen possibilities to interact with the visualisations are provided. The tasks that the tree diagram can facilitate are mainly Overview, Decompose and Relate. Depending on the implementation, tasks like Zoom (e.g., fish-eye) and Details-on-demand (see the implementation of Treemap in figure 4.8) can also be supported.

History

The need for functionality like History was not evident until applications that used hypertext started to appear. Before that history was often limited to the ability to undo a performed action, mostly just one step back. Today's applications have widgets that present lists of performed actions and it is possible to select actions that the user wants to undo. See for example the undo functionality in MS Word. Web browsers have both forward and backward buttons, and it is in this environment that History is most needed. Designers of web pages seem to have a preference for putting in links for navigation backwards, which can lead to confusion since these links do not always point to the same page as the browser's backward button would. Links on a web page can be colour coded depending on whether or not the links lead to pages that have been visited, and this is also a kind of history functionality. Frequently, however, designers choose to disable this functionality, possibly due to aesthetic reasons.

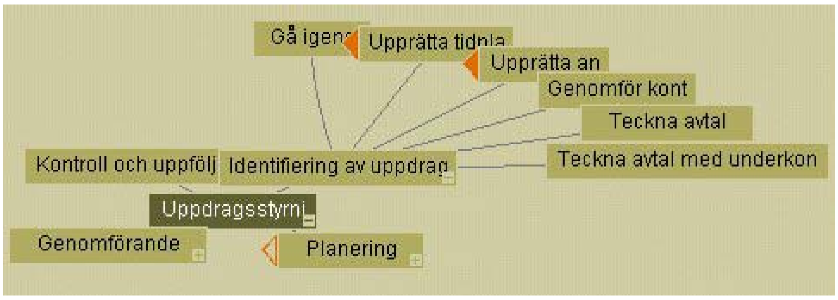


Figure 4.9 Search results marked with a red flag in a hyperbolic browser. Note that collapsed branches that contain the searched word are also marked, but with a hollow flag.

Widgets that effectively visualise history do not really exist. Virtual reality environments have proved to be difficult to navigate through, and research has been done to find ways to facilitate functionality of the same type as history. Extensive process models suffer from problems similar to those in virtual reality environments, and the research performed to make virtual reality models easier to navigate will be valuable to the research dealt with in this thesis.

Search

Search for information can be done in two ways: by key word search or by category. The presentation of the search result is important for the usability of a key word search. Figure 4.9 shows how the result of a key word search in a hyperbolic browser can be presented. It is also important that the user has knowledge about what information is indexed and how to use the search function to limit the number of search hits and to improve their quality. The Internet portal Yahoo, among others, has successfully introduced the category search. The user makes a search by choosing among more and more limited categories and finally gets to a level of detail where the desired information appears. Search by categories is based on a great deal of manual work, since all information that is going to be searchable must be categorised and this work cannot be done automatically by the computer. A computer, on the other hand, can easily produce the index for a key word search.

Details-on-demand

Whenever more information is needed, it should be provided by a Details-on-demand type of functionality. The details should be delivered from the system without being pushy or causing distraction. Figure 4.8 shows an elegant solution. A click on a particular share shows current figures, and another click (not shown in the picture) makes it possible to see a graphic display of the development of the share over time, news items connected to the share, etc. The hyperbolic browser in figure 4.7 can also be connected to external information sources. A double click on a node can, for example, open a web page or a document template that is relevant to the selected node. A characteristic of well-implemented Detail-on-demand solutions is that details are invisible when they are not needed but are readily available as soon as they are needed.

Compare and find patterns

A brilliant example of a well-thought-out tool for making comparisons and finding patterns is the control panel that is part of Smartmoney's implementation of the Treemap, see figure 4.10. The colour key at the top of the panel explains the meaning of the colours. Figure 4.8 shows the Treemap that the control panel belongs to. The pattern that rapidly emerges shows that the Technology and Communications sectors have had a bad day. Today's winners can be found among Energy, Utilities and Basic materials. The changes that are displayed are a comparison with yesterday's closing rates. It is also possible to make comparisons with other occasions. The top five gainers and losers can be highlighted and a search function helps locate specific shares. It is characteristic that functions for searching and making comparisons are integrated in the same control panel. These two functions contribute to understanding the considered system and to finding patterns.

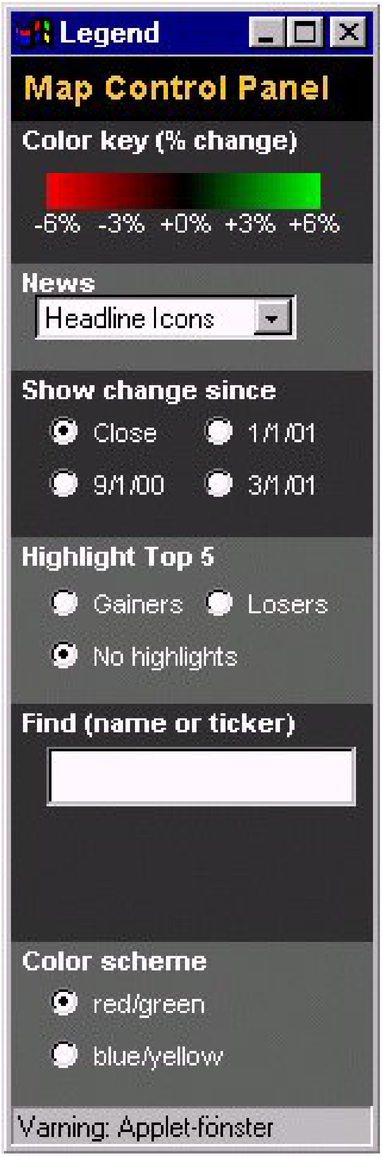


Figure 4.10 Control panel that explains colour coding and manages comparisons of the Treemap in figure 4.8.

4.3 Constructing a process information system

The concepts that have been developed in this chapter can be used to develop user interfaces for process information systems. By sketching a prototype this section shows how this can be done. Some of the functionality described in this discussion is implemented in the prototype that is used in the usability study presented in the next chapter.

The underlying assumption when this system is developed is that an established way of describing processes exists that needs a usable user interface. Any process modelling method can be used, but if a simple box and arrow notation is chosen, there is a risk that the discussion will be trivial. The amount of information and the complexity should be sufficient to motivate and clearly show the strengths of the suggested development method. The IDEF0 process modelling methodology has been chosen in this case as a basis for this prototype. IDEF0 is well documented and has been used by the industry as well as by researchers. The fundamentals of the method are fairly easy to learn. The method is generic in the sense that no limitations concerning type or scope of system exist. However, criticism has been levelled against the diagrams that present the results of modelling according to the IDEF0 method. The diagrams are often considered too information dense and connections between different parts of the model are seen as hard to understand. The study performed by the author of this thesis shows in detail the problems caused by the user interface of this method (Rikard Berg von Linde 2000). In other words, developing a usable user interface that supports this very method is a challenge.

When designing this user interface we start from the generic rule “overview first, zoom in and filter, then details on demand”. Virtual reality modellers often use animations to realise this sequence of activities. Animations do not need to have any active involvement on the user’s part. Hopefully the user will become interested in and enthusiastic about the story that the animation tells. The animation is the equivalent to the introduction that precedes a newspaper article or the anfang that helps the eye find the beginning of a paragraph. A visualisation on an all-embracing level must thus first be created. This visualisation describes the extent and structure of the system being considered. IDEF0 models are hierarchically structured, which implies that some kind of tree visualisation would be well suited as a starting point. Any quantitative information does not usually exist as an integrated part of IDEF0 models, but can nevertheless be linked if required. Starting from an enclosure-type tree diagram is thus not advisable. A node-link diagram is chosen. The nodes will represent sub-processes and the links will represent relations between mother processes and sub-processes. A suggestion of how this can be visualised is given in figure 4.11.

The levels of the model have been emphasised not because they carry any information in themselves but because the marking helps the eye read the diagram the way it is intended to be read. The process is broken down into six sub-processes, which are found on the first depicted level. These sub-processes are in turn broken down into other sub-processes, which are found on the next lower level and so on. Some parts of the process are described in more detail than other parts. The levels are intentionally only faintly outlined and they are, in addition, drawn as billowy curves, which deviates from the rest of the diagram’s design. This is to mark their function as guidelines in contrast to other process information.

Sub-processes are drawn in a way that creates some sort of perspective effects. To move downwards in the model is both a physical action on the screen (or on the paper) as well as a movement

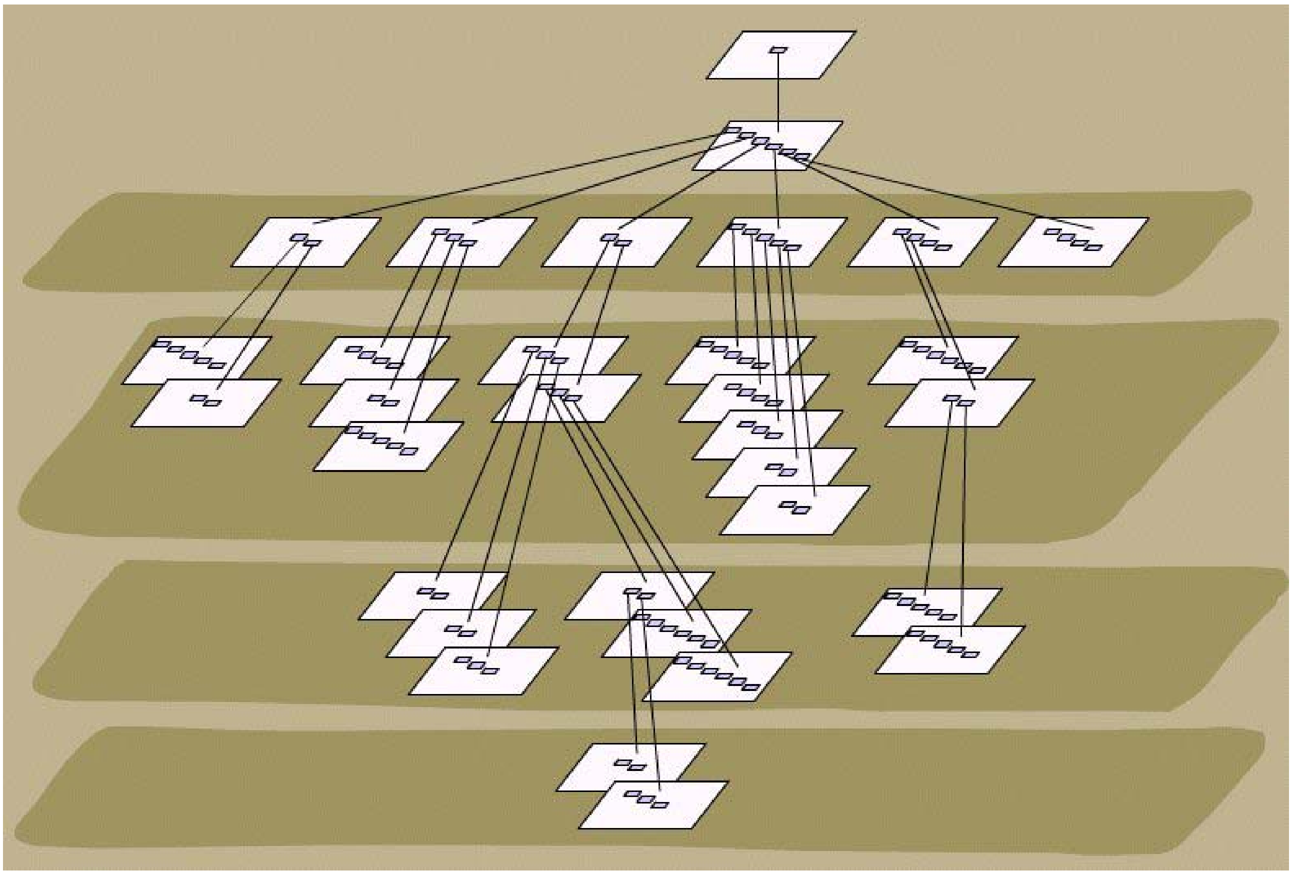


Figure 4.11 Overview first…

down and into the process. The story of the picture explains that by going into a process, sub-processes and their parts will be found. It is possible to amplify the three-dimensionality even more, but there is a risk of adding too much ink to the visualisation that does not carry any information.[3]

This overview picture is the start page of the system. The next step is to zoom in on an appropriate starting point. From this zoomed-in position the continued and more detailed exploration of the model will take place. It is convenient to have the overview picture constantly accessible since it helps to explain the context. The zooming can be done either with or without animation effects. An animation helps the user understand that the original overview picture, the start page, is the same as the miniature that turns up at the top left corner of the window (see figure 4.12). A jump directly from picture one to picture four in the sequence in figure 4.12 may cause confusion. Animations are in some cases laborious to implement technically, but the usability advantages are enough to motivate the needed effort. The position of

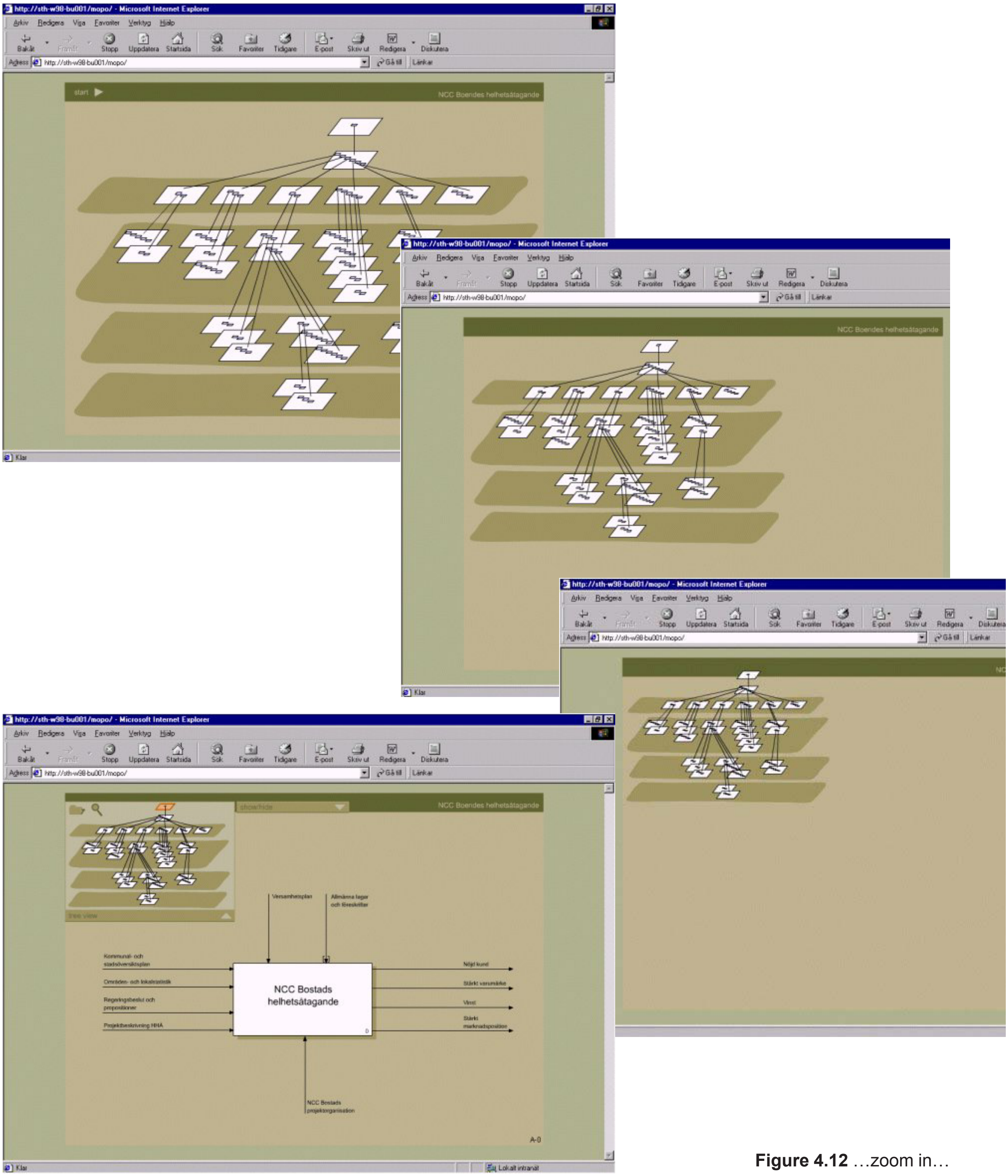


Figure 4.12 …zoom in…

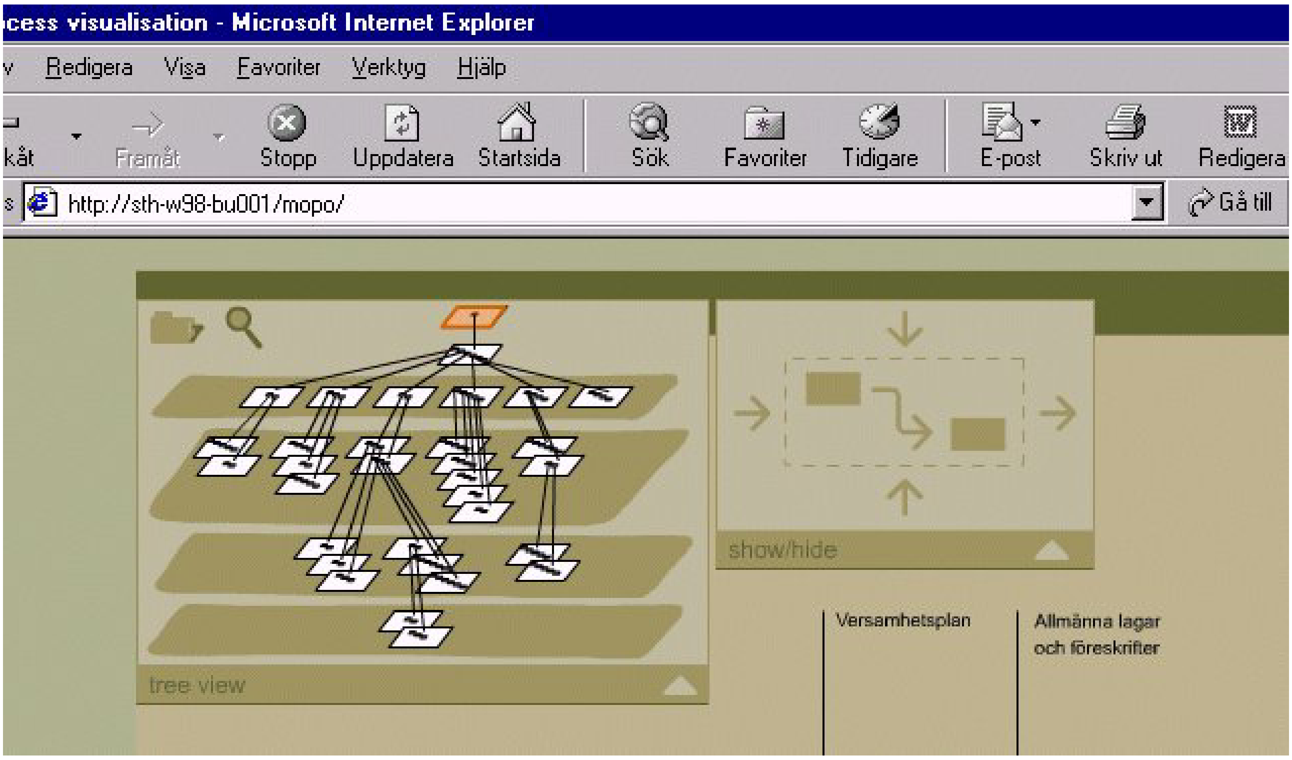


Figure 4.13 …and filter…

the detailed view that is presented after the zooming is highlighted in the miniature overview diagram. The view that is zoomed into is, in this case, given by the IDEF0 methodology. The top box of the model is the only logical starting point in an IDEF0 model.

Overview first and zoom in is now finished. The next step is to filter. When the top box is shown it is possible to either hide details from the beginning (pre-filter) and let the user himself show the objects that are interesting, or as a second alternative, show all objects from the beginning and let the user hide the objects that cover or distract from what interests him (active filtering). The top box of an IDEF0 model is seldom so information dense that the number of objects causes problems. Thus active filtering is desirable on this level. However, all levels below the top box are suitable for pre-filtering since those levels are usually considered too loaded with information. A user interface for controlling filtering of objects in an IDEF0 model can be designed as figure 4.13. By clicking on any of the symbols in the figure, corresponding objects are changed from a visible to an invisible state and vice versa.

The hierarchical structure of the model implies that moving downwards into the model gives an increasingly detailed view of the

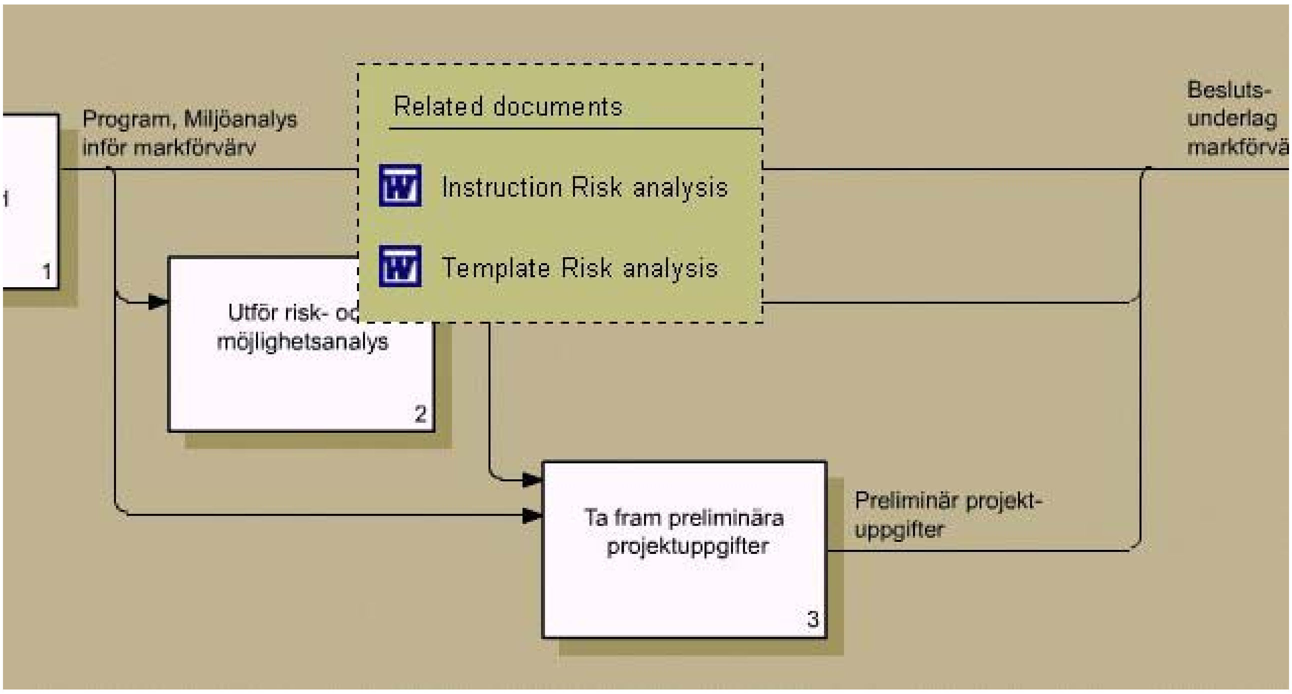


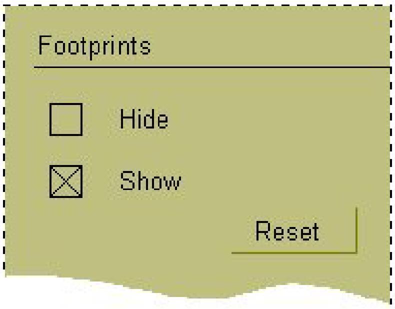
Figure 4.14 …then details on demand. External information sources are connected to sub-processes. In this case it is a detailed description of procedures and a document template that can be used to present the result of a risk analysis.

model. This makes details-on-demand an integrated part of the functions for navigating through the model. Other details can be connected to the model, for example as links from sub-processes to external information sources. If this system is to be used as a basis for a quality management system there would be a need to have more detailed working instructions and document templates linked to the processes. Figure 4.14 shows how this can be implemented in practice.

Decompose is either done by clicking on a process, which will show how this process is split into sub-processes, or by clicking on a process in the reduced overview picture in the top left corner of the window.

Relate is addressed in two levels. The overview picture shows how the parts of the model are related to each other in a hierarchical way. In the individual diagrams it is IDEF0's own notation that indicates the relations between the processes. The IDEF0 methodology describes how arrows are used to connect processes to each other. According to the IDEF0 standard, arrows have different meaning depending on which side of the boxes they join.

History has been discussed earlier in this chapter and it was noted that



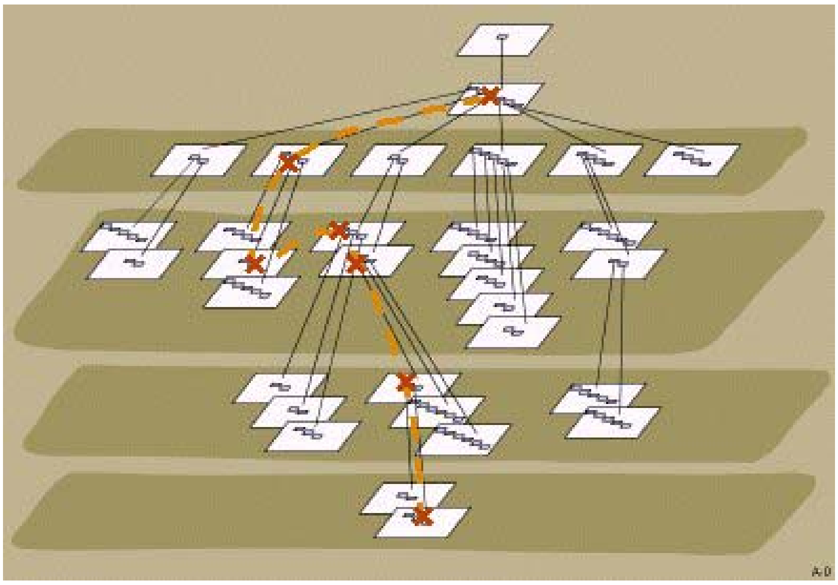


Figure 4.15 The user's movement through the model is displayed in the overview picture. Stops are marked with a red cross.

widgets visualising a user's journey through a system do not exist. However, widgets are available for undoing actions or stepping backwards in a hypertext system. The prototype that is presented in this discussion could make do with these kinds of widgets, but there are other solutions that are more tasteful. The overview picture is suitable for presenting the user's footprints and can in addition be made clickable, which makes it easy to move to any location in the model. Figure 4.15 shows an example of how the electronic footprints can be depicted on the overview picture. An option that resets the register of the user's movements through the model must exist, since the picture will become difficult to read if too many footprints are rendered.

A reasonably easy way of extracting parts of the model is to start from the overview picture and in this picture mark the parts that the user wants to disconnect. The extracted parts form a new model that can be managed in the same way as the original model.

The whole user interface is built up to encourage browsing. No dead-end streets exist in the system. It is always possible to move on either by using the overall picture or the diagrams themselves.

Search is provided as key word search or search by categories (see figure 4.16). The result of a key word search is presented on the overview picture. Some categories of information can be defined, for example, responsible persons. By marking "structural designer" in the

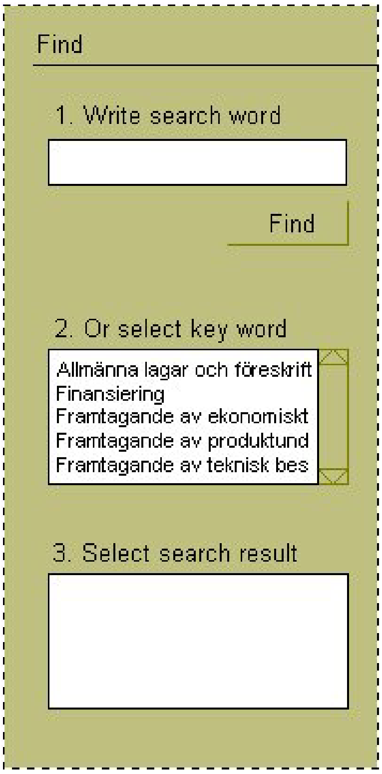


Figure 4.16 Control panel for the search function. The category search is not depicted.

list of areas of responsibilities, all parts of the process involving the structural designer will be highlighted.

By saving the presentation of several search results on the same overview picture, comparisons can be made. For example, it is possible to compare the involvement of different groups in the process.

Find patterns, finally, can be done by using the functions for searching and comparison.

Notes

1. Interactivity can be explained alternately as the work of human and computer application performed to reach an objective. The user asks a question, for example by typing a word into a form and pressing submit. The application performs a pre-programmed action as a response to the user's action. In this thesis, interactivity is also used when discussing paper-based user interfaces. This is not usual, but it is fully logical. A carefully designed book offers many kinds of interactive possibilities to its readers. Consider for example how you use the notes you are reading right now!

2. Node-link diagrams are the generic type of diagram that can be created with connection. There exists no generic type of diagram that can be created with enclosure.

3. Edward Tufte (1983) describes a concept called data-ink-ratio, which he states should be minimised. A diagram visualises data, but to make these data understandable, explanatory text, axes and scale marks are needed. Tufte's rule of thumb is that the amount of ink that is not used to visualise data should be minimised, i.e., the data-ink-ratio should be minimised.

5 Learning from reality

A system is considered to be usable if its intended users can utilise its functions in order to perform predefined tasks. One way to show that a system is usable is to test it on real users, and this is done in this chapter. A prototype designed according to the theory that has been built earlier in this thesis is put into the hands of users – practitioners of the construction industry. It is possible to learn from reality by doing this and to verify that the proposed theory deals with the reality. Another outcome is that deficiencies in the theory can be revealed.

This chapter presents the usability study that has been performed as a part of the research project. The prototype that was used, the test tasks, test persons and procedures are presented. Finally, the results that the study generated are presented. It turns out that the three dimensions, computer practice, domain knowledge and process knowledge, are of conclusive importance in deciding the degree to which the prototype can be considered usable. The implementation of the theoretically identified concepts turns out to be mainly successful, but several users asked for the concepts that were excluded in the prototype.

5.1 The prototype

The previous chapter ended with a description of a system for process information visualisation. The functionality that the system was equipped with was motivated by the list of concepts that generally describes interactive environments for process information.

The prototype that was used in the usability study had less functionality than the system described in chapter 4. This is partly due to limited resources for prototype development, but also to the fact that a more comprehensive user interface can cause results that are difficult to interpret when the usability study is performed. A three-layer model was introduced in chapter 2 that can be used to categorise observations (figure 2.2). The more complex the user interface the more difficult it is to separate users' problems caused by level 1 from problems caused by level 2. It is also hard to distinguish level 2 problems from level 3 problems.

The concepts that were supported by the prototype used in the usability study were the following:

- Overview
- Zoom
- Filter
- Details-on-demand
- Decompose
- Relate

The implementation of those concepts was performed according to the system suggested in chapter 4. The construction company NCC Boende provided the underlying model, as mentioned in chapter 2. The model was structured according to the IDEF0 process modelling methodology.

From a technical perspective, the simplest possible development environment was chosen. The objective was to develop a well functioning user interface as quickly as possible. It was not to create a technical platform that could be developed further, which among other things would entail addressing questions concerning scalability, standardised interfaces (between programs) and documentation. Macromedia Flash turned out to be a suitable development environment considering the objectives. The model that is visualised in the prototype was originally made in the modelling program BPWin. Graphics that BPWin produces could fairly easily be moved over to Flash. The graphics (the diagrams) were disassembled inside Flash and the smaller objects were named to be possible to call with the script language that Flash provides. The script language of Flash is object oriented, but the methods that were developed in this project are too specialised to be of any general value. Despite the fact that the prototype contains a relatively large model, it quickly proved satisfactory and was surprisingly small, approximately 220 kB.[1] Flash uses vector-based graphics, which considerably limits the size of Flash applications.

5.2 Test tasks

A total of nine tasks were prepared. The tasks are presented in table 5.1 along with their originally estimated degree of difficulty. The degree of difficulty was estimated initially, but was revised after test sessions were performed. Some of the test tasks turned out to be much more difficult than expected and were in some cases not sufficiently meaningful to use.

Several of the tasks test the test persons' ability to use the prototype in order to locate certain information. Other tasks require that information be found in the model, but also that conclusions about the information be drawn. Tasks number 1, 2 and 6, for example, test the under-

Table 5.1 Test tasks and their estimated degree of difficulty. The tasks were originally given in Swedish.

Task	Degree of difficulty(1)
1. What activities in the process have strengthened trademark as a result?	1
2. Which professionals are involved in the choice of product?	1
3. On what basis is the risk and opportunity analysis done before land acquisition?	1
4. What activities make use of the risk and opportunity analysis before land acquisition?	2
5. Is it possible to decide on target group before type of housing has been determined?	3
6. What activities generate profit in the process?	1
7. How early (where) in the process is preliminary target group determined?	2
8. When is the final decision on target group taken?	3
9. What does "basis for design decision" consist of?	2

(1) 1 denotes an easy task, 2 a moderately difficult task and 3 a difficult task.

standing of the meaning of input, output, control and mechanism. Tasks number 4, 5 and 9 test understanding of the hierarchical constitution of the model and how different diagram pages are related to each other. Task number 5 also tests the understanding of how succession or sequence is expressed in the prototype. Just locating information has been assessed as the easiest type of task. To locate information and understand its meaning was assessed as more difficult, and to combine information from several diagram pages was assessed as the most difficult type of task.

5.3 Procedures

The usability study was carried out in a standardised way, which is one of several prerequisites for achieving satisfactory reliability. Each test session consisted of six separate steps. These were (1) introduction, (2) check of previous knowledge, (3) training and practice, (4) instructions, (5) performing test tasks and (6) follow up. The objective was that the test sessions should not need more than one hour and 15 minutes to be performed.

(1) Introduction

The study was started with a short presentation of the test administrator and the research project. The importance of the usability study and the test person for the research was explained. The process of the study was briefly and clearly described, i.e., what steps could be expected and how much time the study was going to take. The test person was encouraged to ask questions if anything was unclear.

(2) Check of previous knowledge

Previous knowledge concerning three areas was considered of interest to examine:

Computer experience

Domain knowledge

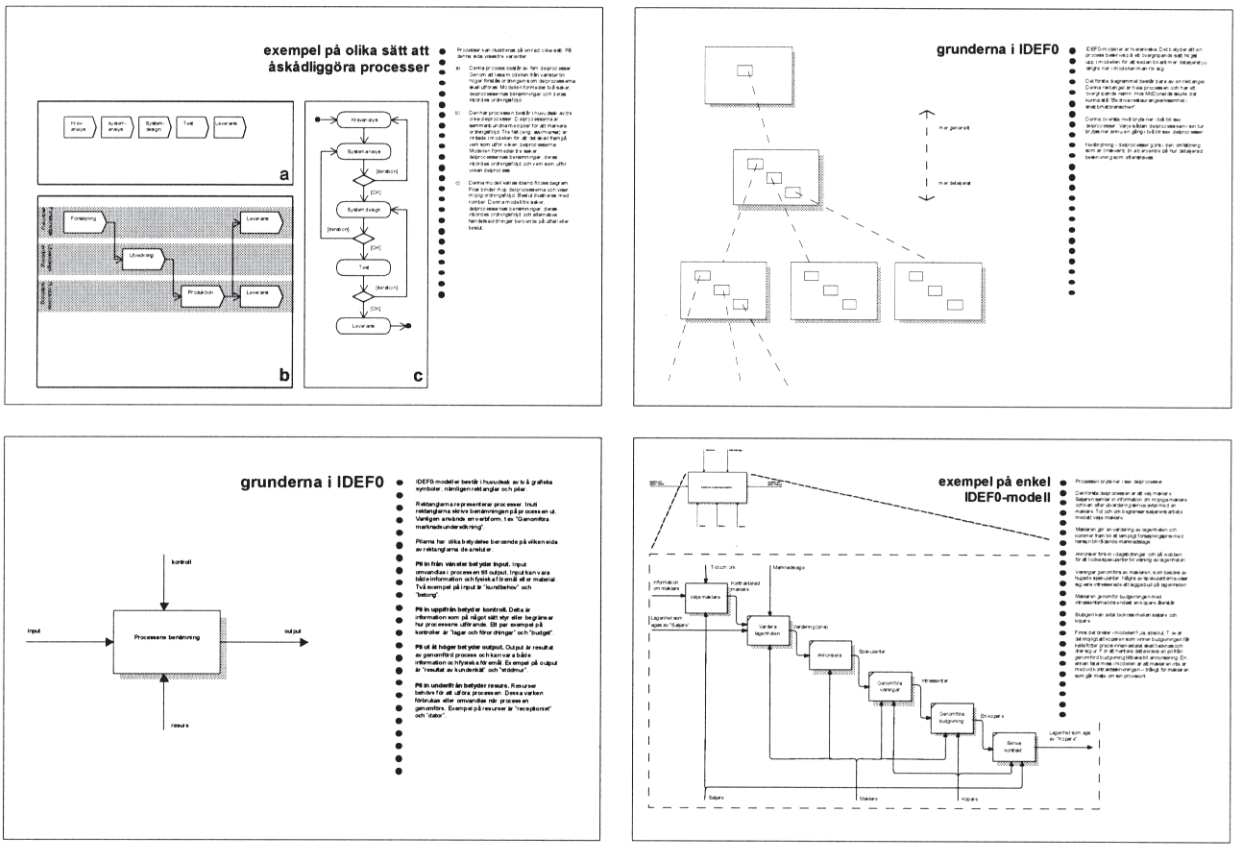
Process knowledge

These three variables were qualitatively examined by asking test persons open questions. Computer experience was examined by asking test persons to describe what computer applications they used on a daily basis and in what way they used them. Domain knowledge was understood by discussing role and assignments of the test persons. Process knowledge referred to general knowledge about processes and process modelling but also specifically about the IDEF0 methodology, since the prototype is to a large extent based on it.

(3) Training and practice

To ensure that the test tasks would be performed in a meaningful way a relatively extensive training effort was needed. Using a set of pictures, the test administrator explained the intended meaning of processes in this context, how they can be illustrated graphically, and how IDEF0 models are interpreted. Finally, an example of an IDEF0 model was shown. Some of the pictures that were used for the training are shown in figure 5.1. At the end of the training session a couple of control questions were asked to verify that the test persons had sufficient knowledge of reading IDEF0 models. The training session took 30 minutes and the pace was adjusted to the test persons' previous knowledge and time needed to assimilate the new knowledge.

Figure 5.1 Pictures that were used for the training. The pictures show examples of ways of visualising processes, the basics of IDEF0 and a simple IDEF0 model that presents the process of selling an apartment.



After this fundamental training session the prototype was introduced. This was also performed with pictures on paper.

(4) Instructions

The test person was instructed on how to carry out the test itself. It was emphasised that the prototype, not the test person, was subject to the test. It was also said that the more problems the test person encountered the more fruitful it would be for the study. There was thus no shame in failing. It was further explained that the test person had the full right to interrupt the test without giving any explanation. How the documentation of the test was going to be performed was also explained, and assurances were made that only the test administrator would have access to the sound recordings. The test person was instructed to think aloud in order to make it possible for the test administrator to understand as much as possible of the test person's actions. The test person was once again encouraged to ask questions if anything was unclear.

(5) Performing the test tasks

The test persons were given time to get acquainted with the user interface of the prototype before they started with the test tasks. Some of the test persons spontaneously tried to use the functions of the prototype. Other test persons needed more direct instructions to get a chance to learn the prototype in practice. After that, the test tasks were handed over one at a time as they were solved. The first task was of the simplest degree of difficulty and the subsequent ones became more and more demanding. Depending on the test person's ability tasks were chosen to be just demanding enough. Test persons who were successful with the initial tasks were given more advanced tasks than those who had more difficulties from the beginning. If problems occurred, the test administrator offered some tips to encourage the test person to continue until the test was completed. How soon the test person stepped in depended partly on the kind of problem and partly on the test person's general ability to carry out the tasks.

(6) Follow up

A follow-up was done immediately after the test tasks were performed. The purpose was to make clear why problems occurred. If there was doubt as to why problems occurred, the task in question was discussed. The follow-up was also an opportunity for the test person to give a critique regarding the application, for example, to tell what features were regarded as useful and what features were missed.

5.4 Test persons

The test persons were selected from among employees of two different companies. One of the companies was a construction contractor and the other was a construction consulting firm. The model presented by the prototype describes a business that belongs to a contractor, but it is also of interest to consultants, for example, structural designers. Among the test persons, several professional roles were represented, for example project managers, structural designers, purchasers, one environmental specialist, quality managers, etc. The level of domain knowledge, computer experience and process knowledge varied greatly among the test persons. In total, 15 persons participated in the study.

Two of the test persons also participated in the earlier study that was performed to investigate the usability of IDEF0 models. These test persons were engaged to give information about how the computer environment changed the experience of working with IDEF0 models. Unfortunately, one year had passed between the previous and the latter study, which may be too long for the test persons to remember how they experienced the last study.

The test persons were asked to reserve 90 minutes for the study. The test administrator visited each test person's place of work and the studies were performed in meeting rooms in close connection to the test person's workroom. The test administrator brought with him everything that was needed for the study.

5.5 The test administrator

The role of the test administrator is worth discussing. In this case the system developer and the test administrator was the same person. There is a risk that persons who evaluate their own systems lack the needed objectivity (Nielsen 1993). This can, for example, lead to the test persons' being offered too much help. It is also possible that a designer will not take user problems seriously enough but will try to find other explanations. This was in fact a problem with the study of this project. The researcher was aware of the problem and tried to make observations while remaining detached. The result became almost the opposite and the self-criticism was probably enough to guarantee a satisfactory degree of objectivity.

The most difficult task for the test administrator during the test is to decide whether it is advisable to interact with the test user or not. As a basic rule there should be as little interaction with the user as possible, but users will run into problems and it is hard to just sit by and watch them struggle and feel unsuccessful. At some point it is clear that the user is stuck and will not manage to get any closer to the solution without help from the test administrator. This is a situation that cannot generate any interesting data at all and therefore it is better to help the user through the situation. In the study this happened for almost all the test persons. Some of the test tasks proved to be unreasonably difficult and had an ambiguous solution (for example task number 5, see table 5.1). Other problems occurred several times and had obvious explanations. In these cases it was decided to help the users early since the problem was already documented and only caused loss of time.

5.6 Results

This is a qualitative study, but in order to get an overview of the result of the study, table 5.2 shows some statistics from the study. The tasks were performed approximately the same number of times, except task number 5, which has been commented on already. About half of the tasks were performed without help from the test administrator. It must be noted that problems that hinder the test person's ability to solve the tasks can belong to any of the three levels of the model for classification of problems (see figure 5.2).

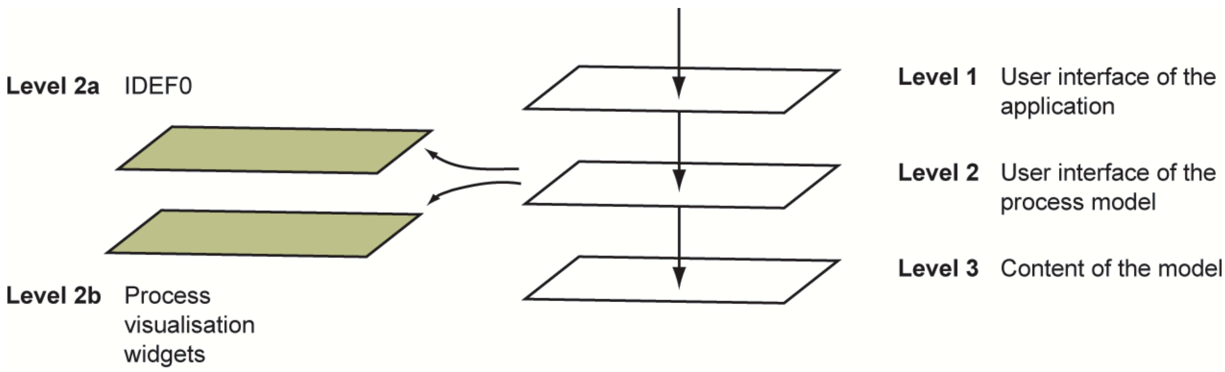
Table 5.2 Results from the user study presented numerically.

Task no.	Degree of difficulty	Times performed	Solved without help
1	1	5	3
2	1	7	4
3	1	6	3
4	2	6	3
5	3	2	0
6	1	6	3
7	2	7	5
8	3	6	4
9	2	5	2

A great deal of the time was spent on giving the test persons the required knowledge about processes, how to read IDEF0 models and how the prototype worked. When the study was planned, this part of it was considered necessary but it was not expected that any useful data could be generated at this stage. But when the study was performed, it turned out that experiences worth presenting were gained. In most cases it is possible to teach how to interpret IDEF0 models in 20 minutes, but the test persons proved to have various levels of understanding; some showed a very shallow understanding, whereas others showed more insight. In connection with the explanation of the basics of IDEF0 a couple of check questions were asked. One of the questions forced the test persons themselves to formulate examples of processes and to decide for those processes what could be input, output, control or mechanism. Persons with different professions and education showed noticeably different levels of ability to define problems and think abstractly. Those who indicated they had problems with the check questions also had greater problems with the test tasks that were given during the actual test session. Experiences from earlier user studies showed that new users have problems understanding how to interpret a sequence from an IDEF0 diagram (Berg von Linde 2000). This was confirmed during the introduction and training phase of the study.

In chapter 2 a three-layer model was introduced to show how observations done in the user study could be derived to different logical parts in the tested system. Already after a couple of studies had been accomplished it was evident that this model was not sufficient to classify the observations. In many cases it was clear what layer an observation could be connected to, and more often than not it was level 2 that caused problems (see figure 2.2). Considering that the purpose of this research is to make process models more usable

Figure 5.2 A more developed model for classification of observations from the user study. Level 2 has been divided into two sub-layers, one for the modelling method and one for the widgets that are available for managing the process model.



principally by removing problems on level 2, this was of course a discouraging result. Worth noticing, however, is that level 2 can be divided into two different parts, one of which cannot be influenced. One of the two parts comprises the design elements that are prescribed by the modelling method used, and the other part comprises the widgets that are available to manage the process model. In figure 5.2 these two levels are named 2a and 2b. The subject of this study is primarily level 2b. Using this more precise model for classification, the results appear less negative and instead a number of improvements can be noted.

The three dimensions that were examined (computer experience, domain knowledge and process knowledge) turned out to be of vital importance for the test persons' ability to solve the tasks properly. The degree of computer experience stood out as having the least influence. Insufficient computer experience should be seen as problems on level 1 and level 2b. Very few problems existed in the study that could be derived to level 1. The problems that were connected to level 1 occurred most often when very experienced computer users were observed. The explanation for this is that the users experienced that there were deviations from what can be called the de facto standard in the prototype. One example is the control panel that is used to show and hide objects on the diagram pages. Tools, which usually have two states in most user interfaces, indicate visually which state exists at the present. The prototype has a mouse-over feature, but there is no indication that the feature is activated. Figure 5.3 illustrates this as well as the wanted functionality.

Earlier, in connection with the discussion of method, it was said that the study would be done on persons with domain knowledge. The degree of domain knowledge turned out to vary considerably among the test persons, which resulted in interesting findings. A test person who is relatively unaware of the visualised process is forced to trust what the process model actually describes. If the process model has a scanty content, this causes problems. Features for key word searching were more likely to be asked for by test persons with poor domain knowledge than by those with good domain knowledge. This was particularly evident when the information that was asked for was found far down in the hierarchy of the model structure. The prototype that was used had no features for key word searching, which made the users step through the model one level at the time. It was not only an advantage to have good domain knowledge. Users who had a strong opinion about how the described process works in reality, or should work in reality, sometimes tried to find information at completely wrong places in the model. Differences between a user's

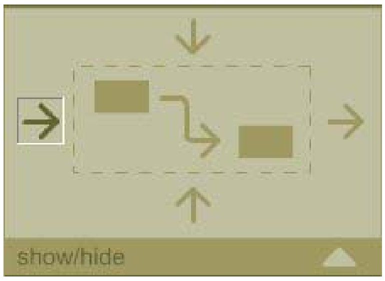


Figure 5.3 The control panel for showing and hiding objects on the diagram pages. When a diagram page is displayed all arrows are automatically made invisible. The arrows can be made visible by clicking the corresponding symbols on the control panel. The prototype missed indication of what objects that had been made visible. The figure shows how this can be done by adding some lines that creates an impression of shadows and that the button is pressed.[2] Several users in the study commented the lack of this functionality.

mental picture of the process and the picture that the model conveyed caused such problems. Problems of this kind occurred in the study, but they should be connected to level 3, the content of the model. Tools that visualise the model should, however, contribute to reducing problems originating from level 3. In this case the search feature that was asked for by test persons with poor domain knowledge can also be useful for persons with good domain knowledge who do not find the information where they expect to find it.

The level of process knowledge turned out to be the dimension that contributed the most to test persons' ability to solve the test tasks, which was not completely unexpected. Also, the process knowledge varied greatly between the test persons. Some of the users had vague conceptions and little experience working with process like information. Time schedules were, for many, the closest they had come to a description of a process. A couple of the test persons distinguished themselves by even being acquainted with the process modelling method IDEF0, i.e., they had seen diagrams drawn according to this method before. These users are to be characterised as having very good process knowledge. Characteristic for the persons with poor process knowledge was that their problems with solving the test tasks in many cases could be derived to level 2a. The basic concepts of how to read an IDEF0 diagram were in many cases not fully understood by those users. This corresponds to the results that come from the paper-based usability study of IDEF0 diagrams (Berg von Linde 2000). The users with good knowledge about how to read process diagrams experienced the tasks as simple and in some cases even as trivial. Some users discovered patterns in the tasks and understood what kind of problem complex the different tasks tested. What is noticeable with those users is that, by controlling levels 1 and 2, they could spend the greater part of their reasoning power on level 3, which is the content of the model. This led to the most penetrating comments about the content of the process model coming from these persons, irrespective of what domain knowledge they possessed.

A clear change of behaviour compared to that in the paper-based study was the way users tackle the tasks. When the model was presented as a pile of papers it caused most of the users to apparently randomly skim through the diagrams on the lookout for a key word that seemed familiar. Only a few users organised the papers systematically and chose to start from the topmost level every time a task was addressed. Most users applied a bottom-up strategy that resulted in poor understanding of the context to which the found information belonged. The prototype led the users to apply a more systematic method to search for information. Almost exclusively, the users worked top-down, which also resulted in a greater awareness of where the user was at any given time. The disadvantage was that experiences were less often gained by mistake. The users seldom happened to display a diagram page; most actions were done on purpose.

Several users showed a great deal of carefulness and did not want to unnecessarily open diagram pages unless they were not sure that the required information was on that page. It was unclear whether the prototype did not encourage browsing enough or if it was the test situation that affected the users to act less spontaneously and in a less well-considered manner.

No history feature, which was asked for by several users, was implemented in the prototype. User studies of general software applications have shown that lack of easy ways to undo actions causes the user to act more deliberately (Dix et al. 1984). The same is true for web sites. The user desists from testing links that are not familiar if there is no easy way of getting back.

A gratifying experience from the user studies was that many users felt that it was stimulating and pleasant to use the prototype. One user said, "How exciting; I can see lots of possibilities". An often-used heuristic for usable user interfaces is that they should be subjectively pleasing (Nielsen 1994), and the study showed that this was fulfilled by the prototype.

To conclude, below is a short presentation of observations from the usability study arranged according to the concepts that the prototype supported.

Overview

Overview was provided by the prototype with a miniature picture of the whole model that was constantly visible in the upper left corner of the application. The miniature picture could be clicked and used for navigation in the model. The feature was appreciated by all users and was used frequently. The users delivered lots of suggestions for improvements and further development. Most common was that the users wanted to have tip text, i.e., when the pointer is held over some of the sub-processes in the overview picture, a tip text that reveals its name is displayed. The toolbars in most Windows applications have this functionality. A possibility to enlarge the overview picture and directly read the names of the sub-processes was also a frequent suggestion for development of the overview feature.

Zoom

The feature that provided zooming was only that the overview picture was possible to click. The user could click on a sub-process and this displayed the diagram in a readable size on the screen. Some users complained that the overview picture was so small that the sub-process was hard to click on with the pointer. The feature was otherwise generally experienced as intuitive and caused no other particular problems.

Filter

When the diagram pages were opened all arrows were made invisible. Several users questioned the usefulness of filtering but often changed their minds after they had spent some time working with the model. One user with experience reading IDEF0 models on paper pointed out that this was one of the features that above all contributed to making the computer-based model more usable than the paper-based model. It was not uncommon that a user had to be reminded that the arrows could be made visible, but after a comment from the test administrator this was generally remembered. Showing and hiding the sub-components of the model, filtering, was experienced as intuitive and straightforward. A wanted improvement was a possibility to show or hide all details on a diagram page with a single click.

Details-on-demand

The arrows of the model denoting input, output, control and mechanism were the only details that could be displayed on demand. Several users observed the possibility of connecting further information to the model, for example as described in chapter 4.

Decompose

Several users had problems understanding the decompose feature of the application. It turned out to be important to have a basic understanding of how a model can be broken down into more and more detailed descriptions, which is the concept of decomposition. Even if the users proved during the introduction and training phase of the test session that they understood what decomposition is, they could still come to completely wrong conclusions when using the prototype. In the prototype, only the overview picture contributed to making the decomposition of the model understandable, which apparently was not enough for all users. Users who had a solid basic understanding of the modelling method managed decomposition in a completely natural and logically correct way. One user who had worked extensively with time scheduling was of the opinion that decomposition (and also sequence in complex processes) should be easy to grasp for persons with experience in construction projects, since time schedules already use decomposition to organise information.

Relate

Relate was shown in two ways by the prototype. At first the overview picture displayed the relationship between the different levels of the model. The other way of showing relations was IDEF0's use of arrows that display the relations between sub-processes internally on diagram pages. In conformity with decomposition it is largely a question of understanding the basic concepts of the modelling method. If this understanding was missing, problems occurred quickly for the users.

Several users clicked on the arrows in the diagram pages, an action which had not been dedicated to any function. When the users were asked what they expected to happen when they clicked on the arrows they did not have any answer. One of the users explained the behaviour by referring to what he called the usual "click disease"; if something looks clickable it must be clicked. Another user thought that clicking an arrow should display the diagram that the arrow leads to. It is difficult, however, to implement this functionality since output arrows, for instance, can lead to multiple sub-processes and diagram pages and it is in that case not obvious what information should be displayed by the system.

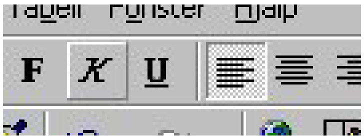


Figure 5.4 Detail from a Windows user interface. The function for justifying text to the left is selected and the pointer (not visible in the figure) is held over the tool for italic text (the letter K stands for kursiv, italic in Swedish). Note how shadow effects create the impression of buttons that are pressed or ready to be pressed and that the light source seems to be situated in the upper left corner of the picture.

Notes

1. The prototype is not a stand-alone application, but needs Macromedia Flash Player to be executed. Web browsers can use a plug-in from Macromedia to run Flash files and this motivates the need for a compact format. It is, however, possible to make stand-alone applications with Flash. Such an application would in this case have an approximate size of 600 kB.

2. The standard of computer user interfaces is that the light comes in from the upper left corner of the screen. A raised button is visualised with a dark shadow drawn at the bottom and to the right edge of the button, and a lowered (a pressed button) vice versa, see figure 5.4.

6 Discussion and conclusions

Two specific topics are of interest for discussion. The first topic is about the concepts. They have been given a principal part in this thesis and thus it is important to critically discuss and examine their value.

The second topic concerns the other essential part of this research, and that is usability testing. Presented results of the usability study appear to be positive and the prototype can thus be considered usable. But there are pitfalls and the answer is far from unambiguous. If the objective is to test users' ability to achieve understanding it is necessary that a model really be able to facilitate the process of understanding – and this can be questioned. What understanding is and models' ability to support understanding are briefly discussed in this chapter.

The conclusions prsented in this chapter are expressed in a condensed form, merely as a summary, since the prior discussion has already presented a more elaborate picture.

Suggestions for further research end the chapter.

6.1 Discussion

You may ask yourself if making process models usable for practitioners of the construction industry is as meaningful as designing a cockpit in such a way that any passenger can manage to use it. There are pilots who are veritable virtuosos in their working environment. The ordinary traveller is completely satisfied to be moved comfortably between different localities. Companies and projects have their own pilots. In contrast to the pilot and his passengers, the communication between people managing the business and the other employees is much more precise and intense. Some company pilots are skilled at analysing the business and its processes. This specialist often makes the analysis alone. It is no use to show the models to the rest of the company as it would be too laborious to explain how to interpret them.

"If you would control the minds of men either deny them information or set them afloat in information – the end is the same."[1] In other words, accept that the models cannot be presented to everyone, or else flood the recipients with information. Neither option is preferable. This thesis tried to create a third option. The two most important means for this was establishment of concepts and the performance of a usability study.

The concepts

It has been argued that a world of concepts is needed in order to discuss this problem in a meaningful way. Regarding communication with process models as a usability problem is not an approach that developers of theories or methods for process management have adopted. In addition, this thesis makes further demarcations of the problem and chooses to focus on the interface between the process model and the user. It is not surprising that accepted concepts that describe this in detail do not exist. Establishing the concepts is thus a result in itself. There are quality aspects that must be regarded concerning these concepts. Most important is probably the question of whether or not the concepts deal with the reality.

The suggested concepts are derived mainly from the information visualisation area combined with experiences from graphical process modelling. The created list adequately describes actions that a user of a user interface for a graphical process model needs to perform. The objects that the graphical process information consists of, however, are faintly suggested. By developing a prototype starting from these concepts an opportunity to test the applicability of the concepts in reality is obtained. The concepts that were realised in the prototype were carefully selected. If all concepts had been implemented in the prototype there was a risk that the application could have been too complex and confounding effects and problems on level 1 could have dominated. The fewest possible, most needed concepts were implemented instead. When the usability studies were performed, omitting functionality turned out to be a successful way of generating reactions from the test persons. Implemented concepts were generally experienced as intuitive and self-evident. No reaction at all was often the result, which is disappointing when performing a usability study. Concepts not implemented, however, were repeatedly asked for. Concepts that were asked for were History, Search, Compare, Find patterns and also a more developed functionality for Details-on-demand. Before the usability study was performed it was assumed that concepts that were not implemented could not be evaluated nor even verified as necessary. Practical limitations that controlled the experiment proved that this was wrong. The result from the usability study can be summarised to state that the identified concepts are useable to describe a user interface for graphical process information. However, the possibility of extending the list further cannot be excluded.

The objects represented by the process information are not investigated in depth. IDEF0 was chosen as the basis for the process information in the prototype. By doing this, it was not necessary to define the objects, but their definitions were given by the standard that describes the IDEF0 methodology. Process, input, output, control and mechanism are predefined concepts and they already have graphical representations regulated by the standard. If another methodology had been chosen these objects would have had a completely different appearance. In the chapter that presents the state of the art it is concluded that the awareness of the importance of graphical design of process models is less well developed amongst theorists and practitioners working with process modelling. Therefore it would be valuable to develop further the concepts that describe the objects. One way of doing this is to start from analysis of what information process models manage to contain. Vesa Karhu has described six different process modelling methods in the form of conceptual models in EXPRESS (Karhu 2000). Described methods are scheduling method, simple flow method, IDEF0, IDEF0v, IDEF3 and Petri Nets. Karhu combines these conceptual models and creates a method that he calls GEPM.

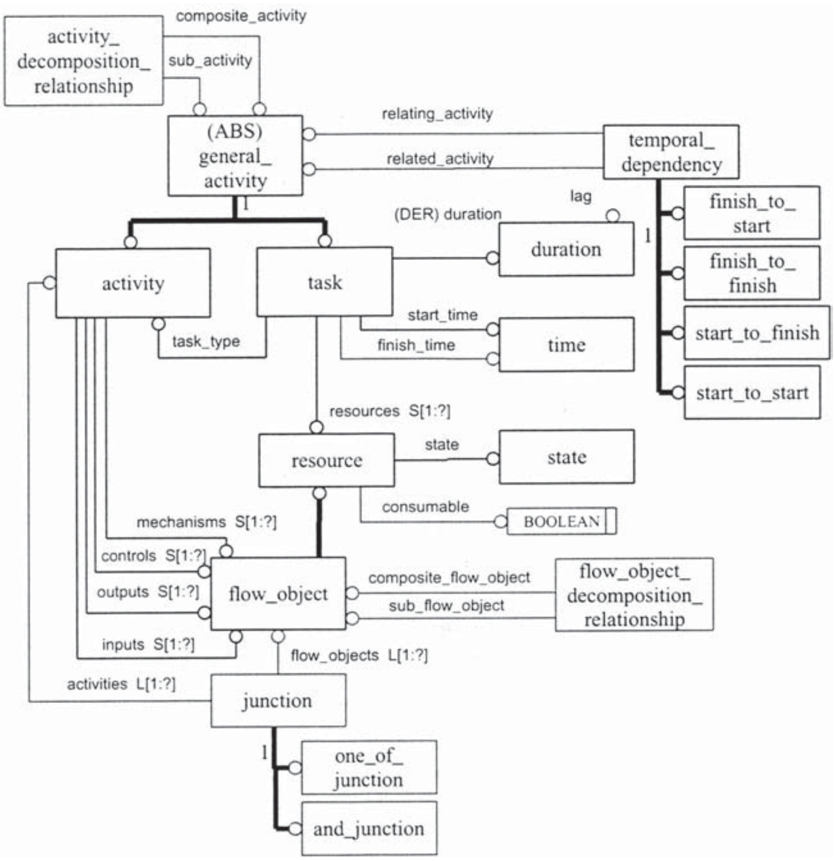
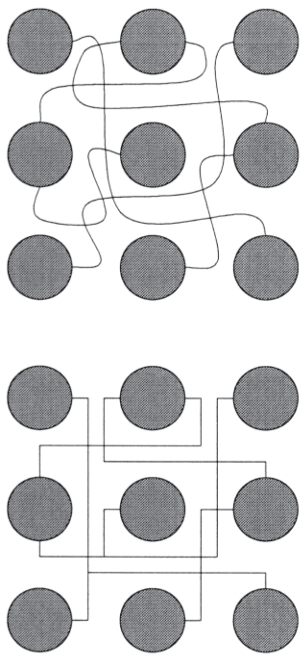


Figure 6.1 Conceptual model of the GEPM method. (Karhu 2000)

Figure 6.1 depicts the conceptual model of the GEPM method. In the conceptual model of GEPM a number of objects can be identified which could be added to the list of concepts.



But what value does the list of concepts have; what is it really good for? In chapter 4 the concepts were used as a checklist when designing an interactive system for visualisation of graphical process information. The text does not tell the whole story. It is much more demanding to design a user interface than just to follow a number of concepts. Merely knowing that history functionality is needed does not help implement the functionality in practice. There are innumerable design decisions that have to be taken before a widget finds its final design.

Every major field of human activity is a mix of art and science (Raskin 2000). It would be presumptuous to believe that it is possible to design a user interface without any feeling for the artistic or acknowledgement of dexterity. The Gestalt School of Psychology was mentioned

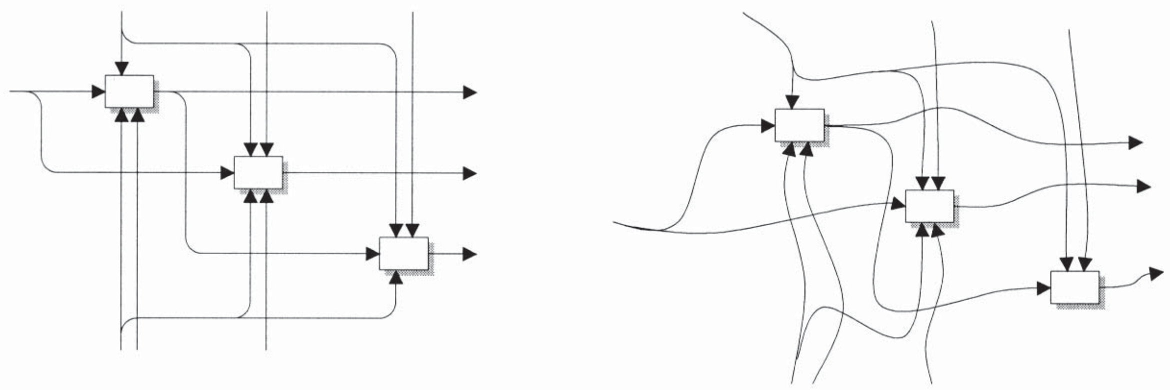


Figure 6.3 An ordinary IDEF0 diagram and a diagram drawn according to the Gestalt law of continuity. It can be argued that the lines are provocatively drawn, it does not have to look this wild, but that is not the point. The point is that a rule of continuity cannot be mechanically used. Art is as important as science.

in the state of the art chapter. It was argued that the guidelines that prescribe how IDEF0 diagrams should be drawn can be explained by referring to some of the Gestalt laws. The Gestalt law of continuity is illustrated in figure 6.2 and a literal application on IDEF0 diagrams can make them completely unreadable. Figure 6.3 shows an ordinary IDEF0 diagram and the same diagram designed more freely but according to the principle of continuity.

Testing usability

The prototype was produced to make it possible to perform a usability study. The question concerns what we learned from the study and whether the prototype really was usable or not. We start by examining the second question.

Methodology issues regarding the usability study have already been discussed, both questions concerning validity and reliability. That discussion stated what is needed to obtain theoretically reliable answers from the usability study. After performing the study it can be noted that no significant deviations from the planned procedures occurred. Accordingly, a positive test result should imply that the prototype really is usable. This merits a closer look. The result of the study was really positive. All test persons did not manage to solve the tasks without problem, but moderately simple instructions were needed to guide everyone to the right solution. The tasks were made to test the user's understanding of the process model. The users proved that they understood by showing that they managed to give correct answers to the tasks. This does not imply that the same user would be able to solve a self-defined problem in his normal work with assistance

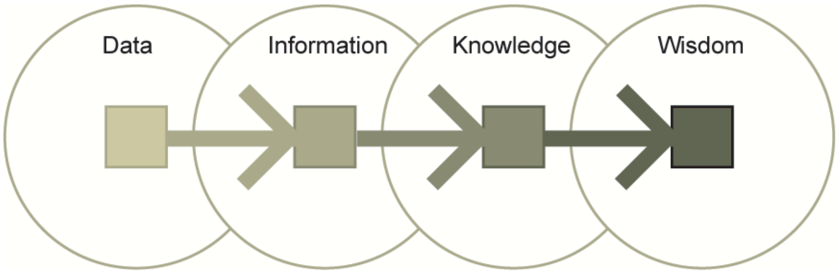
Figure 6.2 The Gestalt law of continuity.

from this tool. The question of whether the prototype can be regarded as an external aid, a thing that makes us smart, still needs an answer.[2] To ascertain this, a more resource-demanding qualitative study needs to be performed. The usage of a fully implemented system would be examined in such a user study, and the users would be exposed in their normal working environment to elements of time pressure and disturbances. Since interesting systems that are already implemented exist, such a study is quite possible to carry out.[3]

However, it would not be incorrect to say that the prototype is usable, referring to the result of the usability study. Understanding was successfully achieved, but not in its deepest sense. Understanding can be thought of as a continuum from data to wisdom (Wurman 2001). This can be expressed as a graphical model, see figure 6.4. The differences between the steps are diffuse and not very easy to explain. Likewise, the concepts themselves are hard to capture, especially knowledge and wisdom. This is due to the fact that at the end of the spectrum, understanding becomes increasingly personal until it is so intimate that it cannot truly be shared with others (Wurman 2001). It can be argued that the process that leads to knowledge and wisdom can be shared. Stimulating thoughts, storytelling, contemplation, interpretation, etc. do this. The truth is probably that a graphical process model will never be able to help a user all the way from data to wisdom. The usability study proved that understanding was obtained up to the second level of this model. It cannot be claimed that the prototype contributed to understanding further along the spectrum.

On some occasions during the usability study something interesting did happen, however. The users started to consider how the work was done in reality and could see discrepancies when comparing the description of the model and the experiences they had themselves. It was not evident that the description that the model represented was wrong, rather, the contrary. The users were of the opinion that it was worth looking over how the work is done today. The conclusion from this is that process models can make users consider analyses they would not have done otherwise, and to discover circumstances that have not been observed before. If this comes into the picture, process models can to some degree facilitate the process that leads to knowledge and wisdom. Common to those users who started to think in this way was that all of them had good knowledge about the domain that the process model described and they also had good knowledge about process modelling in general. Intellectual effort did not have to be spent on understanding things on levels one or two (according to the model presented in connection with the presentation of the results of

Figure 6.4 Understanding as a continuum from data to wisdom. Revised from Wurman (2001).



the usability study in chapter 5, see also the model on page 83), but the users could concentrate on level three, the meaning of the model.

The other question that was asked concerned what we learned from the usability study. An important lesson from the study was that a usability study is an effective way of generating a lot of valuable data without too much time and effort. After relatively few test sessions it was recognised that little new information was discovered and the picture was growing more and more concordant. The greatest variations were caused by differences among the test persons with respect to previous knowledge. It is thus important to collect data on the test persons' previous knowledge about areas relevant to the study.

Another lesson is that the results can be difficult to interpret if no analysis of what to measure has been made in advance. This is because valuable observations risk not being documented in a useful way. However, careful planning can lead to other important issues being missed because the test administrator is not open to other possibilities. A way of compensating for that is to complement the usability study with open interviews and to consider the collected data as being as unprejudiced as possible.

The most positive experience from the usability studies was that the studies were considered instructive and amusing – both by the test administrator and the test persons.

6.2 Conclusions

The most important finding of this research is the concepts that describe an interactive environment for process information. The concepts are grouped into two categories, one for the objects or pieces of information that make up the process model, and another category for the actions that can be done on the objects. The concepts of actions are the most versatile. They can, for example, be used for development of process information systems or evaluation of existing systems for process visualisation.

Some of the concepts were implemented in a prototype and the implementation as such showed that the concepts were practicable. The usability study that was performed on the prototype confirmed this conclusion.

Conclusions can also be drawn about the research methodology itself. A usability study is an effective way of collecting a lot of relevant data. Experiences from working with both simple systems (paper-based user interface) and almost fully developed computer applications show that a less developed system can produce sufficiently good results. If knowledge about usability issues is desired, it is more rewarding to spend time in discussion with users than in bringing a prototype to perfection.

The chapter that presented the state of the art for this research pointed out an important circumstance. It was noted that the knowledge domain of process modelling has shown little if any attention to usability issues. It is easy to find badly designed user interfaces which can jeopardise otherwise sensible process initiatives. A new mixture of knowledge domains can take the use of graphical process models further, not ending at the phase of analysis done by process professionals.

Further research

There are a couple of particularly tempting proposals for further research. The first one is about investigating how graphical process models can be used to facilitate understanding. This issue was touched upon during the user studies when a user suddenly understood that he had experienced something new about the work he thought he knew so well. It is also a question of whether process models are capable of affecting people and organisations.

The second proposal is about finding what a truly usable process model should support. Process models describe changes of state in a system. A construction project is a constant change of state. The predictability is just enough to make it possible to keep the final cost under control and finish approximately at the right time. How can we graphically display something that constantly changes? Putting a video camera on the site monitors the physical activities, but what about the intangible reality? A truly usable visualisation of a process makes it possible to move seamlessly between planned reality and actual reality. Construction managers are incomparable improvisers, and a usable process model must recognise and amplify this ability.

Notes

1. The quotation is from Richard Saul Wurman's book *Information Anxiety 2* (Wurman 2001). In this book, understanding information is discussed in an enthusiastic way with arguments flavoured with notions of democracy and everyman's right to understand the world in order to influence its development.

2. "Things that makes us smart" is an expression coined by Norman (1998).

3. In chapter 1 several examples of process visualisations were presented. One example was taken from a Swedish company called AP Fastigheter. This company has implemented an intranet solution that could be the subject for the suggested study.

References

ACM SIGCHI (Association for Computing Machinery Special Interest Group on Computer-Human Interaction), 2001. http://www.acm.org/sigchi/cdg/cdg2.html.

Berg von Linde, R. 2000. A usability study of a construction related IDEF0 model. Proceedings of the third international conference on product and process modelling in the building and related industries, 25 – 27 September, Lisbon, Portugal.

Board, C. 1967. Maps as models. In Chorley, R. J. and Haggett, P. (Eds.), Models in geography, pp. 671-725, Methuen, London.

Card, S. K., Mackinlay, J. D and Shneiderman, B. 1999. Readings in information visualization: using vision to think. Morgan Kaufman Publishers, San Francisco.

Davenport, T. H. 1993. Process innovation: reengineering work through information technology. Harvard Business School Press, Boston, Massachusetts.

Dix, A. J., Finlay, J. E., Abowd, G. D. and Beale, R. 1998. Human-computer interaction. Prentice Hall Europe, Hertfordshire.

Eriksson, H-E and Penker, M. 2000. Business modeling with UML: Business patterns at work. John Wiley & Sons, New York.

Feldman, C. G. 1998. The practical guide to business process reengineering using IDEF0. Dorset House Publishing, New York.

Gustafsson, L., Lanshammar, H. and Sandblad, B, 1982. System och model: en introduktion till systemanalysen. Studentlitteratur, Lund.

Harris, F. and McCaffer, R. 1995. Modern construction management. Blackwell Science, Oxford.

Herman, I., Melancon, G, and Marshall, M. S. 2000. Graph visualization and navigation in information visualization: a survey. IEEE Transactions on visualization and computer graphics, vol 6, no 1, pp. 24-44.

ISO 9000:2000, 2000. Quality management systems - Fundamentals and vocabulary (ISO 9000:2000). SIS Förlag, Stockholm.

Ivergård, T. 1982. Information Ergonomics. Studentlitteratur, Lund.

Johnson, B. and Shneiderman, B. 1991. Tree-Maps: A space-filling approach to the visualization of hierarchical information structures. Proceedings of IEEE Information Visualisation '91, 275-282.

Kagioglou, M., Cooper, R., Aouad, G., Hinks, J., Sexton, M. and Sheath, D. M. 1998. Process protocol: A generic guide to the design and construction. University of Salford.

Karhu, V. 2000. Proposed new method for construction process modelling. International Journal of Computer Integrated Design and Construction, Vol. 2, No. 3, pp.166-182.

Keijer, U., Svensson, K. and Tarandi, V. 1994. Byggproduktmodeller: Grundläggande begrepp.

Kjaer Jensen, M. 1995. Kvalitativa metoder för samhällsvetare. Studentlitteratur, Lund.

Kolácný, A. 1969. Cartographic information – a fundamental concept and term in modern cartography. Cartographic Journal, 6, 47-49.

Lamping, J., Rao., R., and Pirolli, P. 1995. A Focus + Context Technique Based on Hyperbolic Geometry for Visualization Large Hierarchies. Proceedings of the ACM CHI '95 Conference, May 1995, Denver, USA.

Lester, M. 2001. A picture's worth a thousand words? Available at http://commfaculty.fullerton.edu /lester/writings/letters.html

Lynch, P. J. and Horton, S. 1999. Web style guide : Basic design principles for creating web sites. Yale University Press, New Haven.

MacEachren, A. M. 1995. How Maps Work. The Guilford Press, New York.

Merriam Webster. 1995. Merriam Webster's Collegiate Dictionary.

Miller, J A., Sheth, A. P. and Kochut, K. J. 1999. Perspectives in Modelling: Simulation, Database, and Workflow. In Chen (Eds.): Conceptual Modelling, LNCS 1565, pp. 154-167. Springer-Verlag, Berlin.

Nielsen, J. 1993. Usability Engineering. Morgan Kaufmann Publishers, San Francisco.

Nielsen, J. and Mack, R. L. 1994. Usability inspection methods. John Wiley & Sons, New York.

Nielsen, J. 2000. Designing web usability: The practice of simplicity. Macmillan Computer Publishing.

NIST. 1993. Integration Definition for Function Modeling (IDEF0). Federal Information Processing Standards Publication 183 (FIPSPUB 183). The National Technical Information Service, U.S. Department of Commerce, Springfield.

Norman, D. A. 1998. The design of everyday things. The MIT Press, London, England.

Rao, R. and Card, S. K. 1994. The table lens: Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. Proceedings CHI'94, 318-322.

Raskin, J. 2000. Presenting information. In Information design, edited by Jacobson, R. The MIT Press, Cambridge.

Redmond-Pyle, D. and Moore, A. 1995. Graphical user interface design and evaluation: a practical process. Hertfordshire: Prentice Hall.

Rentzhog, O. 1998. Processorientering: En grund för morgondagens arbete. Studentlitteratur, Lund.

Shneiderman, B. 1998. Designing the user interface: strategies for effective human-computer interaction. Addison Wesley Longman, New York

Tufte, E. 1983. The Visual Display of Quantitative Information. Graphic Press, Cheshire.

Tufte, E. 1990. Envisioning Information. Graphic Press, Cheshire.

Tufte, E. 1997. Visual Explanations. Graphic Press, Cheshire.

Ware, C. 2000. Information visualization: Design for perception. Morgan Kaufmann Publishers, San Francisco.

Wenell, T. 2001. Wenell om project. Uppsala Publishing House, Uppsala.

Woodson, W. E., Tillman, B. and Tillman, P. 1992. Human factors design handbook: Information and guidelines for the design of systems, facilities, equipment, and products for human use. McGraw-Hill, New York.

Wurman, R. S. 2001. Information anxiety 2. QUE, Indianapolis, Indiana.

List of figures

List of tables