Mainport Planning Suite

Software services to support mainport planning

Roy Chin

Mainport Planning Suite

Software services to support mainport planning

Proefschrift

ter verkrijging van de graad van doctor

aan de Technische Universiteit Delft,

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

voorzitter van het College voor Promoties,

in het openbaar te verdedigen op 12 november 2007 om 12:30 uur

door

Roy Ted-Hung CHIN

ingenieur luchtvaart en ruimtevaart

geboren te Leiderdorp.

Dit proefschrift is goedgekeurd door de promotoren
Prof.dr. H.G. Sol en Prof. dr. ir. A. Verbraeck

Samenstelling promotiecommissie:

Rector Magnificus	Voorzitter
Prof. dr. H.G. Sol	Technische Universiteit Delft, promotor
Prof. dr. ir. A. Verbraeck	Technische Universiteit Delft, promotor
Prof. dr. W.E. Walker	Technische Universiteit Delft
Prof. mr. dr. J.A. de Bruijn	Technische Universiteit Delft
Prof. dr. W.G. Vree	Technische Universiteit Delft
Dr. S.C.M. Geertman	Universiteit Utrecht

I wish I could change big things - we all wish that. But at least I can change some little things. The big puzzles are built with small pieces.

Shakira Isabel Mebarak Ripoll

Colophon

Published and distributed by:

Roy T. H. Chin Thomas Jeffersonlaan 387 2285AN Rijswijk The Netherlands Web: www.roychin.nl E-mail: info@roychin.nl	Delft University of Technology Faculty of Technology, Policy and Management Jaffalaan 5 2628BX Delft, The Netherlands Phone: +31-15-2788380 Fax: +31-15-2783429

English editor: Miranda Aldham-Breary
Printing: Gildeprint, Enschede
Cover design: Roy T.H. Chin
Quotation on the previous page: Mercury News, January 29, 2003

Mainport Planning Suite: software services to support mainport planning.
Doctoral Dissertation, Delft University of Technology, The Netherlands
ISBN: 978-90-5638-177-6

Keywords: mainport, planning, decision support, visualization, web portal

Preface

Sustainable growth and the commercial success of "Mainport Holland", located in one of Europe's most densely populated areas, is threatened by a lack of available land, a congested infrastructure, and an increasingly complex social, economic and political reality. To deal with these threats mainports, such as the Port of Rotterdam, are reengineering their planning processes. Instead of making plans based on an extrapolation of current trends, the aim is now to find answers to what-if questions which are applied to concurrent scenarios. Mainport planning is like solving a large jigsaw puzzle, but unlike a jigsaw puzzle the pieces used to solve the puzzle are not available beforehand, and there is not a single best solution. Solving the mainport planning puzzle is a difficult, lengthy, knowledge and information intensive, multi-actor process. In the research presented in this thesis, we introduced the design of a Mainport Planning Suite, MPS, a suite of software services to support the actors in a studio-based planning process and improve their effectiveness in mainport planning.

In this research I was helped by many colleagues, professionals, family and friends. I would like to thank everybody who contributed to this research. I want to thank my supervisor Henk Sol for keeping me focused and guiding me towards my goal: this thesis. I also want to thank my second supervisor Alexander Verbraeck for showing me the many ways that would lead me to my goal.

Many thanks go to my colleagues at the Systems Engineering section who made my time at the TU Delft a rich, happy, and multi-cultural experience that I will never forget. I want to thank Niels Lang and Uta Kohse who stimulated me to start my Ph.D. research. I want to thank Peter Jacobs for improving my Java skills. I want to thank Mariëlle den Hengst-Bruggeling and Wieke Bockstael-Blok for helping me with the evaluation of my research. I want to thank Sabrina Rodriques for her valuable administrative support. I want to thank Elisangela Kanacilo for being a nice officemate and for reviving my interest in South American music. Furthermore, several people helped me in improving this thesis. I want to thank Stijn-Pieter van Houten, Edwin Valentin and Marijn Janssen for providing valuable feedback. Furthermore, I want to thank Miranda Aldham-Breary for improving the quality of the English in this thesis.

This research was conducted in close cooperation with the Port of Rotterdam (PoR), for which I am very grateful. I want to thank Joop Smits, Jan-Willem Weststrate, Priscilla Veenstra, Cleo Hupkes-Lenger, Anca Verbeek, Jeroen Wortel, Frits Apeldoorn, Linda Ristic, Erik Schalkwijk and all the other people at the PoR who contributed to this research. I want to thank Lydia Stougie, Robert van der Linden and Rob Stikkelman from the Port Research Centre, Rotterdam Delft. I also want to thank Kishenkumar Bhaggan who saved me a lot of time finding out how Geotools works. Furthermore I want to thank the researchers who participated in the ABS-project and especially Warren Walker, Jurek Keur, Dries Visser, Roland Wijnen and Paul Roling. I also want to thank Simon Taylor, Steven Turner, Navonil Mustafee, Chen Dan, Wang Xiaoguang and Song Ying for the great time in Singapore. Finally, I want to thank to my paranymphs and brothers Patrick and Ron Chin.

Last but not least, I want to thank you, the reader of this thesis. This thesis would be worth nothing if it remains on a bookshelf. I sincerely hope that this research helps you to create the pieces you need to make a difference.

Roy T.H. Chin

Contents

1 Effective mainport planning

1.1 Introduction

The Netherlands has been known as a trading nation. The nation acts as a gate to the European hinterland, especially because of its unique situation at the European river delta. Next to being a trading nation, The Netherlands is increasingly important as a knowledge economy. Due to increasing competition from cheap labor in developing countries, The Netherlands is shifting from agriculture, fishing and manufacturing towards an ever more service-oriented society. To maintain a sustainable position as a leading trading nation, The Netherlands must continuously invest in its knowledge and innovation industries, which in turn is expected to lead to new products and services[1].

The Dutch mainports, the Port of Rotterdam (PoR) and Amsterdam Airport Schiphol (AAS), are both very important for the development of trade and knowledge (Gout et al. 1997, Kuipers 1999). The PoR is the largest port in Europe and the seventh largest port in the Word, while AAS is the fourth largest hub airport in Europe in terms of passenger movements. Both organizations have a profound influence on Dutch industry, which often requires specific knowledge and continuous innovation to stay ahead of competition. Examples of such industries are container transshipment, petrochemical installations, distribution and logistics, automation of baggage handling and resource planning. Furthermore the European hinterland is reached using an extensive and complex infrastructure consisting of roads, rails, waterways, airways and pipelines. These established trade routes invite multinationals to locate their headquarters and sales offices in the vicinity of the Dutch mainports, which in turn triggers a chain reaction of activity in these areas. Hence, the sphere of influence surpasses the corporate boundaries of the port and airport.

Mainport Holland, as we will call the sphere of influence of the Dutch mainports, is highly interrelated with the Randstad: a continuous chain of urban development including cities such as Amsterdam, The Hague, Rotterdam and Utrecht. This interrelatedness has positive results (employment) and negative results (noise, pollution, safety). Sustainable growth and the commercial success of the recently privatized Mainport Holland, located in one of Europe's most densely populated areas, is threatened by a lack of available land, a congested infrastructure, and an increasingly complex social, economical and political reality.

The planning of, for example, a fifth runway at AAS, or the Second Maasvlakte at the PoR is an extremely complex process, that takes many years, involves many different stakeholders with different objectives, and gives rise to an abundance of information. In contrast to Homo universalis, or Renaissance Man, a modern intellectual can not strive to know all, but has to be content with specializing in a specific field of knowledge. Seen in the perspective of bounded rationality, it is simply impossible to become the Da Vinci of our modern age. Knowledge and information are decentralized, distributed over many people, departments, organizations and nations around the world. Modern information and communication

[1] The Dutch government has made investing in innovation, knowledge and education a priority to stimulate the economy, see www.regering.nl (2006).

technology is used to facilitate the transfer of information and knowledge[2], most notably by the rise of the Internet in the 1990s. However, having access to more information does not necessarily lead to more knowledgeable decisions or more effective planning.

In this research we focus on the challenge of effective mainport planning, specifically focusing on Mainport Holland as our application domain. We investigate a new paradigm for information systems that can be used to enhance decision-making. As introduced by Keen and Sol, the next generation of decision support technology should leverage the knowledge of domain experts and specialists to the executive level, enabling executive managers to "rehearse" the future (Keen and Sol 2007).

We first address the question of what is meant by the term planning in section 1.2. Next we introduce simulation as the preferred method of inquiry to support mainport planning in section 1.3. In section 1.4 we describe our findings on the next generation of decision support technology, which are used to support our research question in section 1.5. Finally we describe our research approach and give a research outline in respectively section 1.6 and section 1.7.

1.2 Planning

Mainport projects such as the expansion of Schiphol in the 1990s take many years of planning (De Vrede et al. 1993, Stumpe 1997, Gout et al. 1997). During this planning period dozens of reports are written by research institutes, consultancy firms, universities, working groups and governments. These reports describe the current situation, regulations, boundary conditions, developments and problems in the mainport region. Furthermore they introduce alternative solutions, evaluations of alternative solutions, sensitivity analysis, and motivations for making a choice between the available alternatives. Many different aspects are addressed by experts from different disciplines such as economics, environment and pollution, traffic and transportation, infrastructure, sustainability, investment costs, and employment. Furthermore mainport planning deals with scenarios and trends. Borrone (in Gout et al. 1997) states: "for a successful long term policy it's also important to anticipate trends in the market". The future is in motion. Certain developments can be extrapolated using relevant data while for other developments it is uncertain what the future will bring. In the latter case different scenarios, such as a worst and best case scenario, can be analyzed.

Clearly mainport planning has many facets, but to start our challenge let us first consider the theory of planning in general. What is planning? What is needed for effective planning? Simon (1986) explains that planning is strongly intertwined with problem solving and decision making. Problem solving deals mainly with fixing agendas, setting goals and designing actions, while decision making is usually focused on evaluating and making choices. During a planning process, alternative solutions are searched for and deliberate choices are made between these solutions. In this sense solving problems and making decisions are part of a planning process. However, it is also possible to regard planning as part of a problem solving or decision making process, because the outcome of planning is a plan used to support e.g. a decision making process. In this section we aim to clarify what we mean

[2] The transfer of knowledge happens in the form of information, however information does not contain knowledge. As Churchman (1971) explains: "to conceive of knowledge as a collection of information seems to rob the concept of all of its life".

by planning, its relation to problem solving and decision making, and the consequences that these distinctions have for this research.

According to Simon (1983) several distinct ways of representing and thinking about problem solving tasks have emerged: problem solving as search, as reasoning and as constraint satisfaction. He argues that in no sense these metaphors are exclusive, but depending on the problem domain a certain viewpoint may be more adequate or efficient. Simon (1986) argues that empirical research shows that: “people solve problems by selective, heuristic search through large problem spaces and large data bases, using means-ends analysis as a principal technique for guiding search”.

Simon (1977) argues that planning can be explained as a form of heuristic search, which involves carrying out the search using a tree model (or directed graph). The tree model consists of nodes which represent states of affairs, and branches which represent the operations that transform one state into another. The sequence of operations which connects the starting node with a given goal node is called a process description. Simon (1977) defines problem solving as: “the activity that seeks to find the mapping between state descriptions and process descriptions of desired situations”. Planning is to find a path from the starting node to the goal node, in what Simon calls abstract space (Figure 1). The original problem space (tree) is mapped to abstract space, by defining sub problems. Each sub problem is a node in the tree representing abstract space. Simon argues that the search for a problem solution in abstract space can be interpreted as a search for information that can be used to guide the search in the original tree model.

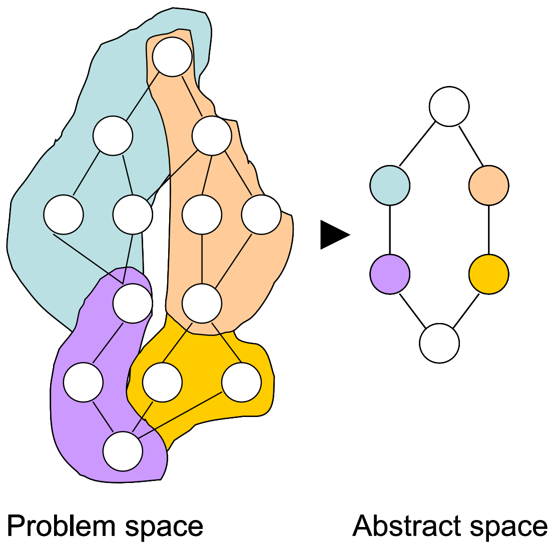


Figure 1: Planning is a search in abstract space to guide a search in the original problem space.

In human planning processes, however, the tree is usually not explicitly defined. In the paradigm of bounded rationality (Simon 1976) humans are not able to see the complete problem space. Besides, in reality no single, static well defined problem exists, instead the problem is continuously changing. Thus, the problem space is always incomplete and inaccurate. Nevertheless the problem space can be large, and as Simon (1977) explains effective planning, and problem solving in general, is largely influenced by the ability to evoke knowledge and information from memory. Memory in this case does not refer to the human ability to remember[3], instead it refers to the information carrier in the broadest sense: human memory, computer memory, corporate databases etc.. As Simon explains the ability to evoke information from memory does not so much refer to having access to (more) information, but merely to extracting the 'right' information. The right information, in this sense, is information which guides a search towards a solution.

Adhering to the paradigm of bounded rationality we argue that a quest for effective planning is not a search for optimal solutions, but a search for acceptable, or satisficing solutions. One, the optimality of a solution is affected by the cost of the problem solving process itself e.g. in time, money, resources. Two, even solving well defined problems is restricted by computational efficiency e.g. the traveling salesman problem. Three, empirical research shows that under uncertainty humans either over-respond or neglect new evidence. Four, and finally, an optimal solution may not exist, as the pay-off of a choice can be a vector of individual preferences (Simon 1955).

In addition, planning concerns the future, an as yet non existent situation. Therefore, a search for solutions is a search in a problem space that is still largely unknown. Adhering to Simon (1977) we conclude that a plan can be seen as a design to reach a desired future state of affairs. In this sense planning has strong similarities with designing. Simon explains designing as reasoning from action to state. The outcome of designing is a design, which if applied in a specified way, is assumed to result in a specified state. Designing is an open process which enables many good solutions. There is no algorithm guiding the problem solver to a solution, instead designing requires creativity to find a solution (Roozenburg and Eekels 1991, Ackoff 1978). Planning as design also reveals the relation between planning and decision making: Simon (1977) explains designing as a decision making activity, leading to alternative solutions among which a deliberate choice can be made.

From our investigation of planning we can conclude that:

- a search for solutions is actually a search for knowledge and information from memory
- the ability to evoke knowledge and information from memory largely influences the effectiveness of planning processes
- knowledge and information are evoked from memory, and the design of alternative solutions also contributes to memory, resulting in an accumulation of information during a planning process
- as planning involves design, it requires creativity to find a solution

The challenge of effective planning now becomes a quest for an effective problem solver.
The challenge is to support invoking the memory and the creativity of multiple actors, with different objectives that are specialized in different fields of knowledge, and that work in different contexts, such that effective planning can be conducted (Figure 2). Our aim is to address this challenge with a new paradigm for decision support. This requires us to first choose the method of inquiry to be supported, which we discuss in the next section.

[3] The Cambridge Advanced Learner's Dictionary defines memory in three ways: (1) the ability to remember information, experiences and people, (2) part of a computer in which information or programs are stored, or (3) the amount of space available on it for storing information.

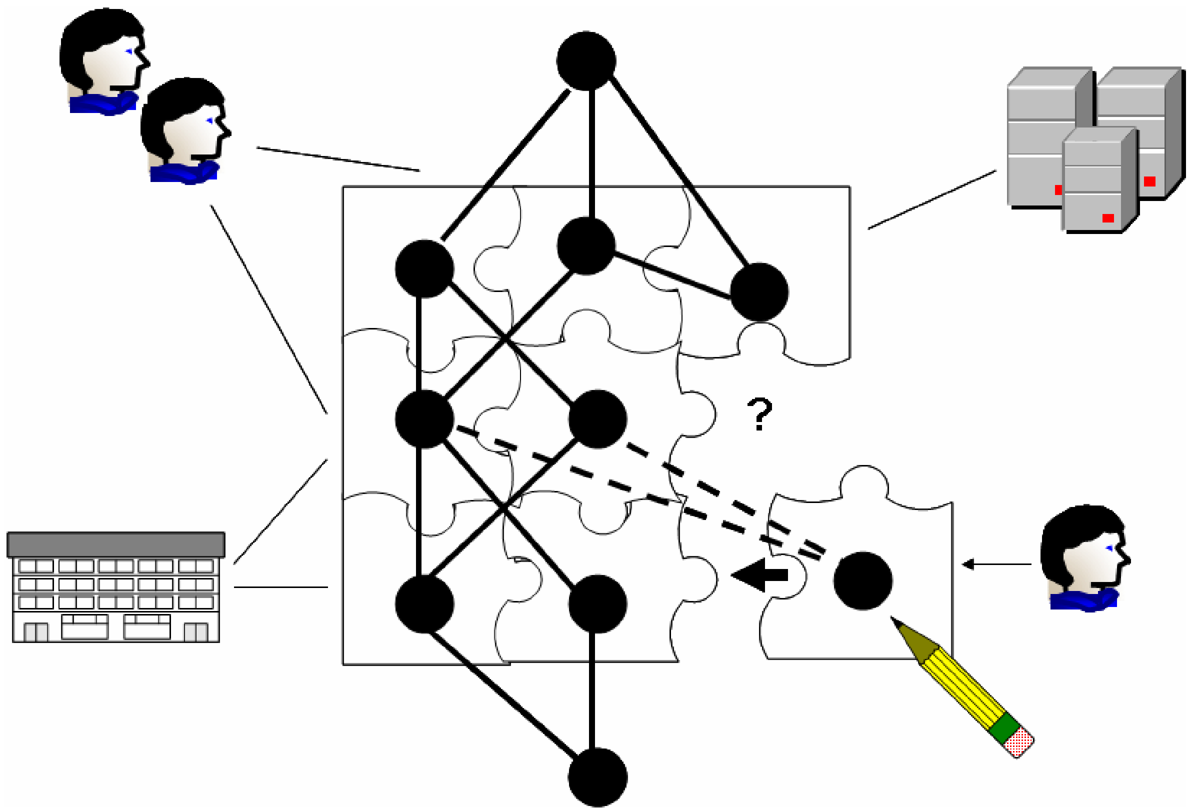


Figure 2: An impression of the "planning puzzle".

1.3 Simulation as a method of inquiry

In the first section of this chapter we argued that today man cannot strive to know all, but has to specialize in a specific field of knowledge. However, as Ackoff[4] argues, specialization has been taken too far by the beginning of the 20th century, with the sciences divided into increasingly narrower disciplines. Using an analogy Ackoff shows that you cannot understand the workings of a system by taking it apart: "a car is not a car anymore when you take out the engine." Vickers[4] clarifies this: "a system is both more and less than the sum of its parts." Such deliberations lead to the development of systems thinking, i.e. seeing the system as a whole in all its technical, social, economic and political complexity.

Mitroff et al. (1974) agree with Ackoff that science had mostly become the product of disciplinary thinking. However, after the World War II a host of inter-disciplines emerged and according to Ackoff: "Science was putting itself back together". Mitroff et al. respond by outlining a program for studying science from a holistic point of view. Their systems view of "different varieties of scientific behavior" is presented in a diagram (Figure 3) which shows the different activities and stages of problem solving.

[4] The BBC Open University provides online video interviews with Ackoff, Vickers and Checkland, explaining their ideas on systems: http://www.open2.net/systems/practice/index.html (last accessed March 2006)

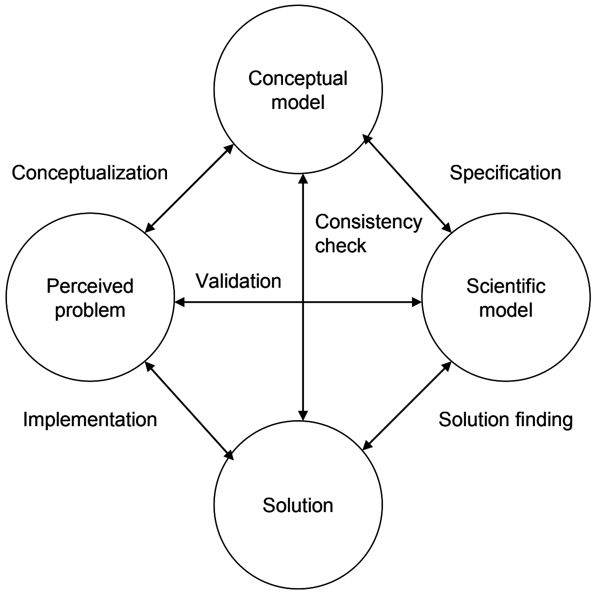


Figure 3: A systems view of problem solving (Mitroff et al. 1974).

Four stages can be distinguished in problem solving: perceived problem, conceptual model, scientific model, also called empirical model, and solution. The activities that can be distinguished are conceptualization, specification, solution finding, implementation, consistency check and validation. Combinations of these activities lead to a *model cycle* (Sol 1982), which represents a process of problem solving. While problem solving often starts with a perceived problem, Mitroff states that any of the stages can be the start point for scientific inquiry, leading to at least four distinct ways in which inquiry can begin. He explains that if different individuals trying to solve different problems in different contexts are compelled to choose different starting points, they are also compelled to choose different end points.

The activities that together form a process of problem solving can be supported by an inquiry system: a structured set of instruments which can be used as a context in the problem solving activities (Sol 1982). Sol's definition of an inquiry system is an extension of Churchman's (1971) definition of an inquiring system. We make a distinction between Sol's and Churchman's system by referring respectively to an inquiry system and an inquiring system. Where Churchman focuses on the activities, Sol adds instruments to support these activities. Churchman investigates the extent to which man can design an inquiring system. He defines inquiry as an activity which produces knowledge. Churchman describes a classification of systems, showing an evolution from primitive to more advanced forms of inquiring systems. He advocates the last, and thus most advanced system described for solving ill-structured problems: the Singerian inquiring system. A Singerian inquiring system uses the "whole scope" of inquiry, meaning that in principle all disciplines can be used in problem solving. Hence, a Singerian inquiring system supports an interdisciplinary approach, and where the problem solver seeks to solve a problem by querying it from many different viewpoints including the scientific, ethical and aesthetic.

Sol (1982) presents simulation as a Singerian inquiry system, and thereby advocates simulation as a method of inquiry for solving ill-structured problems. Traditionally simulation is defined by Shannon (1975) as: "the process of designing a model of a real system and conducting experiments with this model for the purpose of either understanding the behavior of the system or of evaluating various strategies for its operation." Furthermore Zeigler (1976) provides an extensive overview of the field of modeling and simulation and the instruments used in simulation experiments. Adhering to Sol (1982), simulation can be regarded as a method of problem solving. Simulation as a method of inquiry can be expressed in the cycle of dynamic modeling, which describes a specific elaboration of the problem solving cycle described by Mitroff et al, based on the Singerian inquiring system advocated by Churchman (1971) (Figure 4). This elaboration is as follows: the perceived problem is split into a problem situation and a changed situation, making it possible to evaluate the effects of a solution to decide whether or not a new cycle is necessary. Furthermore the scientific model is split into an empirical model and models of alternative solutions. Several actions are specified in this respect: analysis, finding solutions, ex ante evaluation and choice. Together with conceptualization, specification and validation, analysis results in an understanding of the problem situation. Finding solutions is a creative activity, or a design activity leading to alternative solutions. Ex ante evaluation is aimed at gaining insight into the effects of alternative solutions before these are actually implemented. Finally a solution is chosen for implementation.

Following on the deliberations on planning as a human problem solving process presented in the previous section, we choose to adopt simulation as our method of inquiry. Masser and Ottens in (Stillwell et al. 1999), who focus specifically on urban planning, argue that planning is mainly centered on analysis, design and decision making, i.e. evaluating alternative solutions and making a choice for a preferred solution. Furthermore, they stress that these activities are typically performed by multi-disciplinary teams. In line with Masser and Ottens we choose to center our research on the activities of analysis, finding solutions, ex ante evaluation and choice, as indicated in the shaded box in Figure 4.

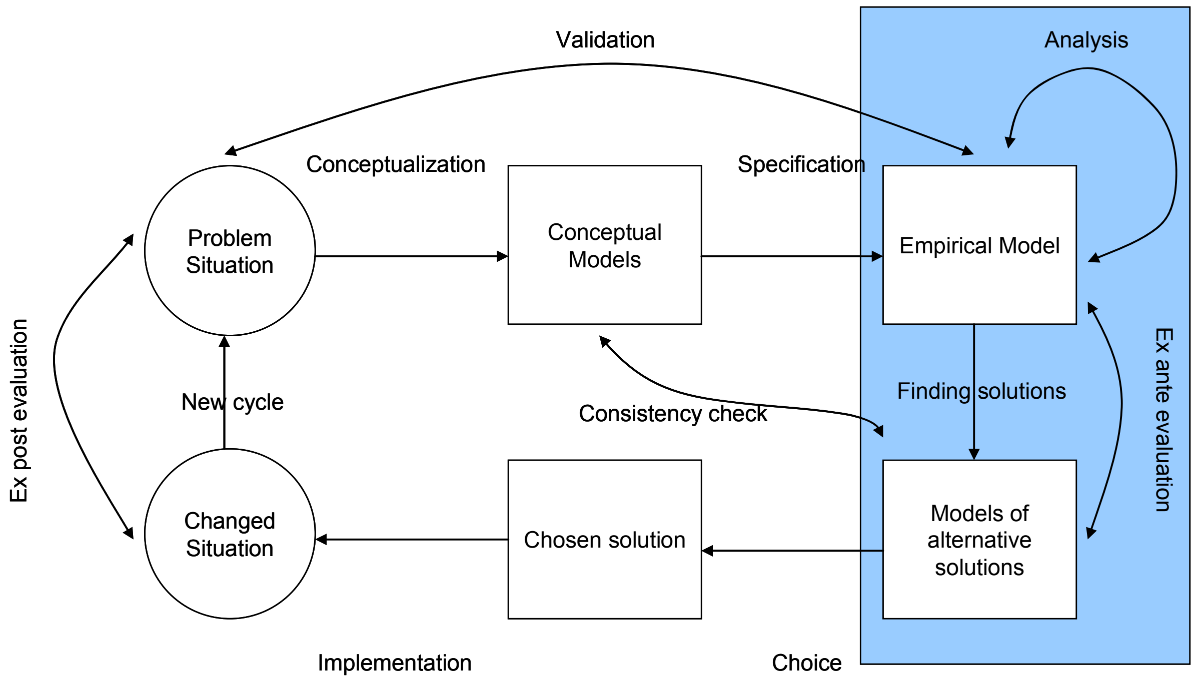


Figure 4: Cycle of dynamic modeling.

1.4 Studio based decision making

We focus on simulation as a method of inquiry, as discussed in the previous section, and following Sol (1982) simulation should also provide the instruments to support the activities of inquiry. Based on the characteristics of mainport planning, as introduced in 1.2, we believe that using the method of simulation to support mainport planning requires simulation instruments which are specifically designed for this domain. For example, in Stillwell et al. (1999) the role of e.g. GIS, visualization systems, and decision support tools in spatial planning are extensively addressed. Following Sol (1982) we conclude that simulation instruments can be used as Decision Support Systems (DSS) (Keen 1987). Carlsson and Turban state that the original promise of DSS was to improve the effectiveness of decion makers in dealing with ill structured problems, to help decision makers in making better and more reasoned decisions, and to support decision makers in making systematic use of their knowledge and information. However, as Carlsson and Turban (2002) conclude the original promises of DSS have not yet materialized. They argue that DSS do not sufficiently address the cognitive constraints of humans. Similarly, Jacobs (2005) argues that currently the instruments of simulation merely support a substantive rational approach to problem solving.

Keen and Sol (2007) recognize that information technology still has little impact on an executives' decision making. There is a huge opportunity gap between the needs of decision-makers and computer- and communication tools. They argue that effective decision support puts an equal emphasis on three Us: usefulness, usability and usage. Usefulness relates to the added value, usability to ease of use, and usage to the flexibility, adaptivity and sustainability of decision support technology.

Keen and Sol (2007), envision a Singerian inquiry system in the broadest sense: the decision support studio. Studios are facilitative simulation environments designed to enable executives to rehearse the future. As they explain: "Rehearsing the future rests on vision, envisioning shared images, collaboration and communication among people scattered across the organization and more and more outside relationships". Studios are not simulation environments in the traditional sense of the word, instead studios "embed system tools in executives' decision processes".

Studio based decision support encompasses:

- people: the actors representing a whole scope of disciplines and inter-disciplines in the context of the problem to be solved
- process: a process founded in the method of simulation, guiding search towards a solution
- technology: suites of interrelated information and communication services, simulation instruments, analytic methods, visualization interfaces

As Keen and Sol (2007) explain, a suite is the foundation for meshing technology and process. A suite consists of domain specific information and communication services, which form building blocks that support recipes for repeatable processes. Jacobs (2005) links the concept of a suite of services to the service oriented computing (SOC) paradigm. In this paradigm a service is defined as a self-describing, open component that supports the rapid, low cost composition of a distributed information system (Papazoglou and Geogakopoulos 2003). In a way a suite is a toolbox for studio based decision making, similarly a plumber has a toolbox that can be used to fix a broken outlet pipe or replace bath taps e.t.c.. A surgeon uses a specific toolbox to operate on a specific patient. Like the surgeon, an executive, planner, or expert often faces different problems which require using a specific set of instruments. The choice of instruments will depend on the context, the roles of the involved actors, and possibly, the personal preferences of the managers involved.

Oosterhuis (2003) introduces a similar concept in the field of architecture. Today architects work in collaborative environments where they are immersed in an overwhelming flow of information. He envisions studios, or “transaction spaces”, where information is exchanged between people, between computers, between people and computers, and even between buildings. The people, computers and buildings are like nodes in a network that receive information, process information and send information in a collaborative design project. The information is stored as data in a distributed “project database” which exists in the architects’ minds, in documents, and on computer hard disks. The actors cannot access the project database directly, instead suites of services form the interface “to make it possible for us to give meaning to the data, to make interpretations, to propose changes, to produce new data and to have them sent back to the database” (Oosterhuis 2003).

1.5 A suite to support mainport planning

As we described in the beginning of this chapter, planning is a search for knowledge and information in a distributed memory, or “project database” as Oosterhuis calls it. An ability to evoke knowledge and information from this memory is a key factor for an effective planning process. However, mainport planning is characterized by a lengthy process, which involves many stakeholders, and many different types of information which have to be considered in parallel. That this is not an easy task was illustrated, for example, by experiences of planners during the expansion of Schiphol in the 1990s (De Vrede et al. 1993, Stumpe 1997, Gout et al. 1997).

Recent developments in the field of decision support and collaborative design and the development of decision support studios, provides us with instruments that can be used to support the method of inquiry. Therefore we have based this research on the hypothesis that suites can be used as an enabling technology to improve the effectiveness of mainport planning.

Following the above, our research question was formulated as follows.

Research question:
Can we create a suite of services to support the actors in a studio-based planning process that improves their effectiveness in mainport planning?

As we will discuss in our research approach, in the next section, this research question was used as a temporary measure as our research strategy allowed it to be sharpened as we proceeded with our research (see section 2.3).

1.6 Research approach

1.6.1 Research philosophy

"It's the question that drives us."[5] In our search for an answer we, mankind, have developed a system for the acquisition of knowledge. Characterized by a systematic process of inquiry, the system aims at discovering, interpreting and revising facts, to obtain an accumulation of knowledge which eventually approaches the ultimate goal: to know the truth. This system of research is called science.

Acton (2004) believes the main question asked by scientists is: "how can we come to know anything about the universe?" How this question is interpreted will vary, depending on the underlying assumptions different paradigms in the philosophy of science represent different approaches. Acton provides a tree of "approaches to knowledge", making a distinction between passivist and activist branches.

Passivists interpret the question as a question of truth. According to passivist paradigms such as those of the inductivists, probabilists and dogmatists, the answer can be found "out there". To a passivist, observation statements are considered to be the foundation of all meaningful concepts. In contrast activist paradigms, such as those of the Kantians and revolutionary conventionalists, interpret the question as a question of understanding. Activists believe that sensory observations are theory laden, and always infected by one's preconceptions. Observations are not objective, or mechanistic but obtained through selection and active acquisition of the mind in the search for knowledge.

As explained by Gregg et al. (2001), in information systems research the passivist and activist paradigms are usually represented by respectively the positivists and the interpretivists. Positivists assume that there is no difference between the real world and the social world. Instead, the analogy they assume is that only one reality exists. Their truth regarding information systems can be described as decontextualized, objective, and quantitative. Interpretivists assume that to understand an information system, one has to "plug in". They view an information system as a social system in which multiple socially constructed realities coexist: only their active participation in the resulting context of "digital life" can provide them with an answer to their mostly qualitative question.

In addition to positivists and the interpretivists, a socio-technologist or developmentalist paradigm can be distinguished in the field of information systems research (Gregg et al. 2001). While the positivist and the interpretivist ask the possibly most pertinent question "what is?", to the developmentalist paradigm this question seems to be irrelevant. The developmentalist does not aim to build knowledge by researching what is true, but by researching what is effective (Hevner et al. 2004). The keyword describing the developmentalist paradigm is "creation". Information systems are regarded as social systems that are technically implemented. However, nothing is perfect, except for maybe nature itself, and therefore any design is fundamentally flawed. The developmentalist depends on positivist and interpretivist inputs for falsification and confirmation, bringing to light any anomaly in design, and eventually to support iterative progression in the design cycle (Vaishnavi and Kuechler 2006). Ergo, the question of the developmentalist is the question of progress, which was also our question for this research.

[5] A wink to the famous Hollywood epos of the Wachowsky brothers: The Matrix.

1.6.2 Research strategy

Our choice for a research strategy was shaped by our research philosophy. March and Smith (1995) introduce design science and explain its difference with regard to natural science. Whereas natural scientists and behavioral scientists try to understand reality, design science attempts to create things that serve human purposes. Natural science often consists of discovery and justification. Design science revolves around building and evaluating objects and systems. Hevner et al. (2004) stress the complimentary nature of both paradigms, as design relies on using existing theories. They state that design science is pro-active with respect to technology, whereas natural sciences, or behavioral sciences, take technology as a given. They strongly advocate a research cycle where design science is used to create artifacts aimed at solving specific information problems, based on relevant theory from the field of natural science. Furthermore, in this research cycle natural science anticipates and engages the created artifacts. We follow Hevner et al. (2004) in their proposed complementary, exploratory research strategy.

Design science products come in four types: constructs, models, methods and instantiation. As March and Smith describe, the constructs form the vocabulary of the research domain. A model is a set of propositions or statements expressing relationships among constructs. A method is a set of steps used to perform a task. An instantiation is the realization of an artifact in its environment.

Three types of research contributions can be distinguished in design science: the design artifact, foundations and methodologies. This research was focused on the production of a design artifact that could be used to facilitate the solution of heretofore unsolved problems. Design is inherently an iterative and incremental activity. In our field the business environment establishes the requirements upon which an evaluation of an artifact is based. Thus an evaluation includes the integration of the artifact within the technical infrastructure of the business environment. The use of prototypes is therefore *necessary* (Hevner et al. 2004). They also stresses that design evaluation should include an assessment of the artifact's style, i.e. "machine beauty".

1.6.3 Instruments

Hevner et al. (2004) propose a framework for information systems research. Due to the iterative and incremental nature of design the evaluation phase is necessary to provide essential feedback to the construction phase. In their framework, Hevner et al. provide an overview of evaluation instruments which must be matched appropriately with the designed artifact and selected evaluation metrics (see Figure 5).

1. Observational	Case Study – Study artifact in depth in business environment
	Field Study – Monitor use of artifact in multiple projects
2. Analytical	Static Analysis – Examine structure of artifact for static qualities (e.g., complexity)
	Architecture Analysis – Study fit of artifact into technical IS architecture
	Optimization – Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behavior
	Dynamic Analysis – Study artifact in use for dynamic qualities (e.g., performance)
3. Experimental	Controlled Experiment – Study artifact in controlled environment for qualities (e.g., usability)
	Simulation – Execute artifact with artificial data
4. Testing	Functional (Black Box) Testing – Execute artifact interfaces to discover failures and identify defects
	Structural (White Box) Testing – Perform coverage testing of some metric (e.g., execution paths) in the artifact implementation
5. Descriptive	Informed Argument – Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artifact's utility
	Scenarios – Construct detailed scenarios around the artifact to demonstrate its utility

Figure 5: Design evaluation methods (Hevner et al. 2004).

We choose to construct our research around literature research and case studies (Yin 1994). Case studies allowed us to study the artifact in the business environment. We made a division between exploratory case studies and case studies for evaluation of our design. The exploratory case studies each essentially represented an entire design cycle, where the falsifications and confirmations found in the former were used to inform the latter, eventually leading to a revision of the initial research question. Each case study was complemented with instruments such as observation, interviews, questionnaires, and testing to challenge the design from both the positivist and the interpretivist perspective. As such we decided to use the following instruments.

- Observation: mainport planning was observed in practice during a case study to provide a better understanding of the mainport planning process.
- Participatory design and workshops were used to encourage the close involvement of the "customer", a representative counterpart from a mainport organization, in the design of our suite. This was expected to result in short feedback loops.
- Demonstration: demonstrating ideas and early prototypes or mockups was used to gain feedback from a larger group of experts in a mainport organization.
- Storyboard, or structured walkthrough: evaluating a prototype of the suite using a structured walkthrough allowed us to focus attention of the participants on those parts of the suite we wanted to evaluate.
- Half open interviews: interviews with individual experts were used to gain a more in depth insight into the qualitative aspects of this research.

- Questionnaires: surveys using questions and statements were used to provide more quantitative feedback.

1.7 Research outline

The outline of this research is shown in Figure 6. Our choice for this outline follows from our research strategy, i.e. design science: our research is characterized by a number of design iterations. Each design iteration represents a cycle of build and evaluate. In chapter 2 we explore mainport planning in practice. We describe two case studies, i.e. our first two iterations, which resulted in a refinement of our initial hypothesis, which we state in the form of a research question. This leads us to the start of our third and last iteration in chapter 3: the introduction of the theory and principles of visualization as a basis for further design. In chapter 4 we combine the functional needs for mainport planning found in chapter 2 and the theory introduced in chapter 3. We select promising technologies to implement the theory and to address the functional needs that were found. The outcome of chapter 4 is a set of requirements for our design. Next, we follow a bottom up approach to describe the design of our suite. In chapter 5 we describe the technical building blocks which form the basis for our design. In chapter 6 we describe how these building blocks are used to design the functional services of the suite. In chapters 7 and 8 we evaluate our hypothesis of effective mainport planning. Finally in chapter 9 we present our conclusions.

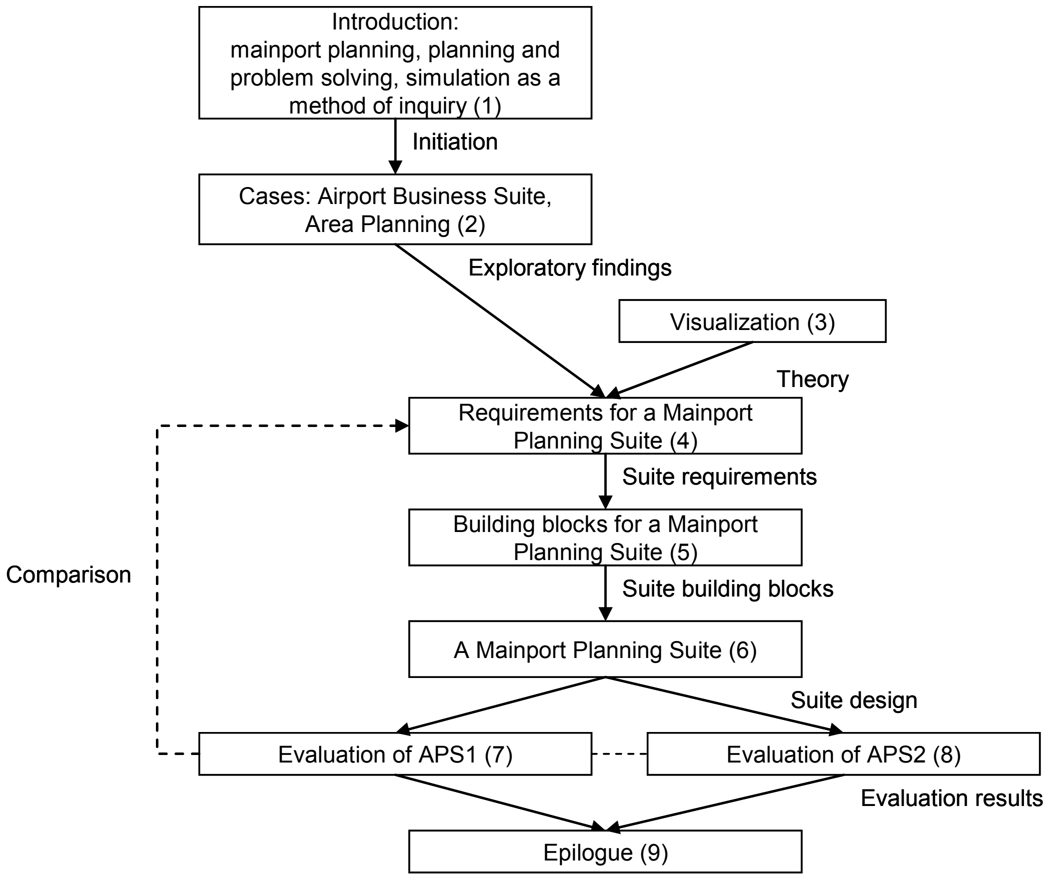


Figure 6: Research outline

2 Supporting mainport planning in practice

In this chapter we present two exploratory case studies. We used these case studies to increase our understanding on supporting mainport planning, and to find out what is needed for a suite to support mainport planning. The case studies were conducted sequentially, and an evaluation of the first case was used as a basis to improve the second case.

2.1 Supporting airport business planning

2.1.1 Introduction

Amsterdam Airport Schiphol (AAS) and other major hub airports around the world are in a period of transition: a transition from a state owned national symbol to a privatized business that must compete in the global marketplace. At the same time their market is rapidly changing in response to the rise of low cost carriers, increasing fuel prizes, security measures, and new regulations. Airports need to be flexible and adaptive to operate successfully in this dynamic, complex and unpredictable environment (Neufville and Odoni 2003).

In this case study we investigated a new studio based planning concept named Airport Strategic Exploration (Walker et al. 2003, Visser et al. 2003). ASE differs from the original concept of airport master planning (FAA) in that the aim is to rehearse the future, to do a what-if analysis by trying out policy options and alternative designs under a variety of future scenarios, rather than to produce a static plan for the next 20 years. ASE is not about preparing the future, as airport authorities used to do, but about being prepared for the future. Being prepared means that airport executives can make well informed decisions that matter in a rapidly changing reality. These decisions concern e.g. adding new runways, terminal expansion, attracting low cost airlines, investments in e-ticketing and other services.

This case was characterized by a strong focus on the means to support ASE. As we described in chapter one, mainport planning involves dealing with an abundance of information drawn from a large variety of disciplines. Working out a single scenario in airport master planning is a lengthy and complex process. Our case study started with the assumption that, to conduct a what-if analysis effectively on multiple alternatives, the different models and tools used by the domain experts should be integrated into a consistent suite of services. Different disciplines cannot be considered in isolation, but any one discipline provides input for the other. For example, economic forecasts are used to determine demand for flights, which in turn influences the demand for airport capacity. This integration of different domain specific models was expected to provide useful results faster and with better consistency.

About fifteen researchers drawn from a variety of disciplines, such as policy analysis, aerospace, ICT and economics, were involved in this case. They cooperated for a period of one year in order to design a suite of services to support ASE: the Airport Business Suite (ABS). The author participated in the case as one of the researchers. This case can be seen as an exploration into studio-based decision making, mainly focusing on providing support for the technical and functional needs of airport planners. Which activities should the suite support? How can information from different domains be integrated? How should the results be presented? These were major questions for this case study.

2.1.2 Airport strategic exploration

We started with an investigation of ASE and the different disciplines involved in airport planning. Several workshops were organized in which the involved researchers shared and explained their views on ASE. Furthermore, a former airport decision advisor for Amsterdam Airport Schiphol actively participated in the team. Most of the team members were domain experts who were new to the concept of ASE and studio based decision making. Furthermore, they had only a limited knowledge of each other's fields of expertise. In addition a number of interviews were planned with airport planning experts at Amsterdam Airport Schiphol, ABN AMRO and the Netherlands Airport Consultants B.V. (Den Hengst et al. 2002). The outcome of these workshops and interviews were: a clarification of the role of the problem owner, an agreement on the stages and activities in the planning process, knowledge of the needs of the intended users of the suite, and the key factors, and their interrelationships, that must be supported by the ABS.

The actors involved in ASE

The problem owner was the airport decision maker, and the ABS user was the airport decision advisor. The role of an airport decision advisor is to develop alternative strategies for the future development of the airport and to communicate the alternatives to the decision maker: the CEO of the airport. These alternative strategies are presented in the form of a strategic plan. An airport organization may have a small team of airport advisors (about 5), and domain experts who develop and evaluate alternative strategies based on developments related to external influences (economic growth), environmental restrictions (noise, pollution), market developments (airlines, destinations), capacity demand (infrastructure, resources), business strategies, investment plans, and organizational development plans. Airport decision advisors depend highly on human judgment, experience and skills. Detailed quantitative analyses e.g. regarding runway capacity or noise are usually delegated to domain experts, who then present the results of an analysis to the airport advisors.

The process of ASE

During the workshops the planning process of ASE was conceived to be a policy analysis process (Miser and Quade 1985), where policy analysis is defined as "the generation and presentation of information such a way as to improve the basis for policy makers to exercise their judgment" (Walker 2000, Walker et al. 2003). In policy analysis, a problem is identified and objectives are specified. Criteria are selected which have to be met in order to meet the objectives. Next an iterative cycle of selecting, analyzing, evaluating and comparing alternatives is entered, which should finally converge to a choice for a preferred solution.

The information needs in ASE

The airport decision advisor and the interviewees were asked for their needs and expectations regarding the type of support which the ABS should offer. With respect to the information needs, an initial number of information types were identified which should be supported: forecasts of demand for flights, airport capacity information, and financial information such as turnover, investments and operational costs. These information types are closely related to aviation regulations, the airport layout and operations. Consequently these types of information should also be considered in ASE. Furthermore, at a later stage, also information regarding noise, pollution and safety could be supported.

The technology needs in ASE

With respect to technology, the ABS was considered to be an organizational decision support suite which needed to be embedded in the planning process and which needed to be compatible with the tools already in use at an airport. The ABS should be a platform which integrates existing analytical and simulation models into a consistent suite of information services. Furthermore the ABS should be focussed on a single airport, including its airlines, and its in- and outbound connection network. The ABS should typically be useful to address questions regarding changes in flight schedules, the impact of a new runway, or the by that time hot topic of building an entirely new airport in the North Sea. The outcome of an analysis was not considered to be most important, rather the insight into and understanding of the sensitivity of the analyzed alternatives was considered to be of primary importance. Finally, the ABS should not be aimed at replacing the judgment of experts, but rather at supporting a visualization of the sensitivity of alternative solutions under the specified scenarios.

The challenge of supporting ASE

The challenge of supporting ASE lies in the translation of the identified information and technology needs into a number of domain specific key factors, and their interrelationships, which should be represented in the ABS. A number of expert workshops were organized to make this translation. This resulted in Figure 7, which shows the input-output relationships between the identified key factors. The development of the world's economic markets is considered to be the main driver for air traffic at a specific airport. The demand for passenger traffic to and from the airport is assumed to depend on the economic situation in the respective origin and destination airports. Depending on an airport's capacity this demand for traffic can be accommodated to a certain extent, allowing an airport to realize a certain level of service. Airport capacity can be broken down into airside capacity and landside capacity. Airside capacity depends on the available infrastructure such as runways and taxiways, air traffic control (regulations), and the actual weather during operations. Landside capacity depends on terminal infrastructure such as check-in desks, available walking surface, baggage reclaim areas, customs areas, and other resources including airport personnel. Realized flights to and from the airport generate income. Airlines operating at an airport pay airport fees. Many airports provide extensive retail services to passengers to provide additional income, these include shopping facilities, hotels and car parking services. Furthermore, business operating at the airports such as shops, travel agencies, hotels etc. also produce real estate income for the airport. Finally, the total turnover is used to cover operational costs and investments in for example infrastructure.

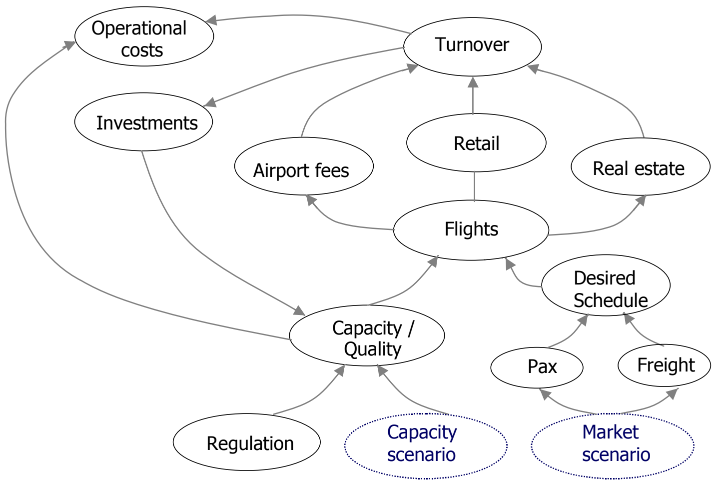


Figure 7: Interrelationships between key business factors (W.E. Walker et al. 2003).

2.1.3 Design

Based on the requirements, the research team made a design of the services to be provided by the ABS. Several workgroups were set to design the ABS. Each workgroup focused on a specific discipline, or model. Furthermore, each workgroup had one member who also served on the integration team, which oversaw the complete development process.

The identified key factors and their interrelationships were grouped into a number of models. A main requirement mentioned in the previous section is that the ABS should make use of tools which were already in use at an airport organization. Consequently the identified models are an example set of the possible models which are in use at an airport. The five models which were identified and their respective input-output relationships are shown in Figure 8. The arrows in the figure should be read as input and output relations between the models. The available capacity and desired schedule provide input for matching demand and supply. This results in a realized flight schedule and passenger flow in the airport terminal, which again is input for the model which calculates airport turnover. Part of the turnover is used to cover the costs of maintaining and running the airport business, and another part is used to invest in new infrastructure, facilities and resources.

Each of the models consists of one or more submodels. For example, ‘supply of airport capacity’ consists of a runway capacity model and a simulation model used to calculate arrival and departure delays (Ball and Swedish. 1981). Each of these models was used to form the basis for a service in the ABS. For example, the service based on the ‘supply of airport capacity’ model provided a graphical user interface to specify the model input data (flight schedule, runway configuration, weather etc.) and visualize the outcomes of running the models (e.g. capacity and delay graphs for arrivals and departures).

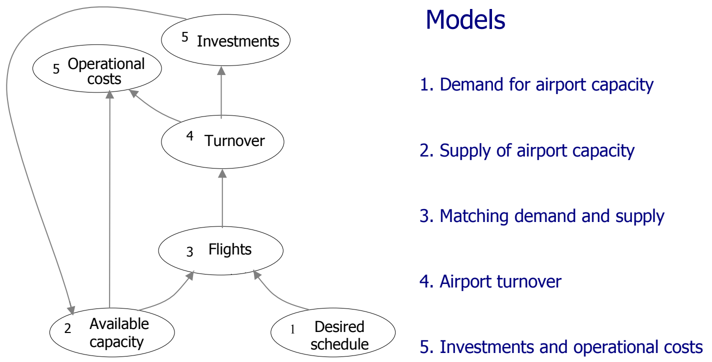


Figure 8: The identified models and their relations (W.E. Walker et al. 2003).

What characterizes the identified models is their difference in level of detail, scope (system boundaries), and time horizon. For example, the *demand for airport capacity* is a model which uses a worldwide scenario for economic growth (or decline) to determine a demand for flights between origin and destination airports for a specified year. However, the input required for the *supply of airport capacity* model is a flight schedule for a single day, because the realized runway capacity depends on air traffic control operations and weather influences. Leveling out capacity over an entire year would not provide a realistic image of actual delay resulting from a mismatch between demand and capacity. In contrast, the realized amount of flights is a measure of turnover, which is determined per year. As a result the models cover a whole range of abstraction levels from strategic to operational. Unfortunately, each step of transferring data from one model to the next involves making assumptions and this leads to accumulating errors. Therefore the ABS cannot be used as a precision instrument, but in line with the identified needs can be used to provide a sense of sensitivity of the outcome when evaluating different policies.

Two units of analysis were chosen, because of the difference in time horizon: the peakday and the year. A peak day represents the X-th busiest day, e.g. the 20th peakday is the twentieth busiest day in the year. The ability of an airport to accommodate traffic on a specified peakday is a measure of its level of service. An airport which is capable of accommodating traffic on the 1st peakday will give an outstanding service, but will have over-capacity during the remaining 364 days of the year. Usually an airport is designed to accommodate e.g. the 20th peakday, so that operations run smooth about 95% of the time.

These two units of analysis are represented in the way experiments are set and organized in the ABS. Each run of a model requires either data at the level of a peak day, or data based on an entire year. Furthermore the outcomes of a peak day analysis can be aggregated to data representing an entire year using a Capacity Coverage Chart (CCC) (Neufville and Odoni 2003).

2.1.4 Evaluation

A prototype of the ABS was implemented to demonstrate its feasibility (Figure 9). The prototype was used for demonstration and feedback purposes. Furthermore the prototype was used as part of a course on airport planning at TU Delft, in which students were required to use real airport data to explore possible expansion scenarios.

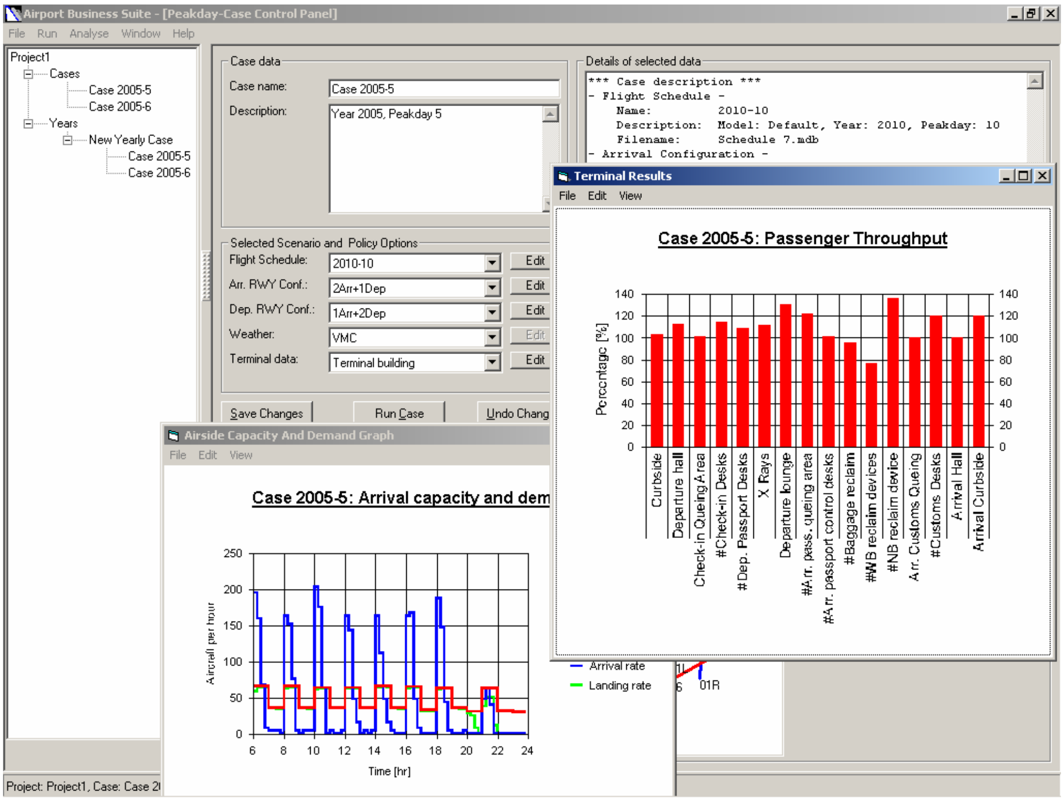


Figure 9: ABS prototype.

The airport advisor who participated in the project had high expectations of the ABS. He actively participated in the development of the prototype and was therefore capable of providing feedback throughout the design and development of the ABS. An airport decision advisor is confronted with an increasingly dynamic, complex and unpredictable environment. He or she cannot predict the future, but aims to get a feeling for the sensitivity of policy options under many possible scenarios. The instruments and tools he requires for a what-if analyis should support him or her in planning and designing in real-time. What if we build that runway over there? What if we delay construction for another year? What if we accommodate this low cost carrier? The airport advisor would like to drag a new runway onto the virtual scene and instantly see the effects of such on arrival routes, noise pollution, safety limits, flight schedules, terminal operations, and airport turnover in real time. Hence, to the airport advisor the ABS should be a system which enables him or her to design or redesign an airport and see the effects in real-time.

For the researchers who participated in the project, the realization of the prototype was a continuous trade off between the wishes of the problem owner and what is technically feasible. Technological complexity was in many ways the limiting factor. For example, the economic forecasting model uses MS Excel, while the runway capacity model is based on a closed source batch program provided by the FAA. Both models require hundreds of input variables ranging from economic growth percentages, aircraft types, and number of passengers to weather conditions, aircraft separation standards, final approach speeds and runway configurations. Furthermore there are differences in detail level, scope and time horizon. For example, economic data are available on a yearly base, while the runway capacity model requires data for a single day and produces outcomes at resolution of 15 minutes. Linking the models required assumptions to be made in the transformation of data from one model to the other. The models do not provide results in the blink of an eye, but may require at least minutes of runtime depending on the complexity of the specified experiment.

Consequently the airport advisor cannot simply drag a new runway onto the scene and see the effects instantly. Assumptions need to be specified, and policy options and scenarios need to be translated into hundreds of model inputs to declare how the departure and arrival routes change, when is which runway used, which peak day is chosen, what is assumed about the weather, how is the mix of aircraft types specified and in which order do they arrive or depart, and so on. These are just a few of the detailed translations which must be made, and after starting the simulation it takes at least a few minutes before results are available. Interpreting such results requires expert knowledge and careful investigation. For example: Why do we see a congestion building up at a certain location in the terminal building now that a runway has been relocated? Is it because of a change in arrival delay, because of assumptions made concerning the wind direction, or is it due to the choice made to use data for a certain peak day?

The ABS does not provide support for the translation of policy options and scenarios into model inputs. The models need reliable input data, and the outcomes of interest cannot be interpreted without expert knowledge. The added value of ABS is mainly to be found in the integration of different models where the output of one model provides input for another model. This automatic transformation of data between models saves time and increases the consistency of input data between models. For example, the outcome of the demand model is a flight schedule which forms the input for the capacity and delay model. After the demand model finishes, the capacity and delay model can immediately use the computed flight schedule. However, the capacity and delay model also needs hundreds of additional input parameters, such as the runway configuration, traffic regulations, and weather conditions. Defining these input parameters requires knowledge from domain experts.

Clearly the airport advisor depends on the knowledge and experience of domain experts. The students at TU Delft who worked with the ABS as part of a lab exercise used to mimic a professional airport planning team. They too divided tasks and each became responsible for a specific domain. A suite like the ABS does not automate the functions of the domain experts, instead it strengthens the need for collaboration and deliberation between different disciplines. Experts in civil engineering, air traffic control, terminal operations, aircraft noise, safety and economics need to facilitate the translation of strategic policies into detailed operational model inputs. They need to cooperate to ensure that they use the same assumptions, boundary conditions, initial values and so forth. Planning a new runway is not a computational exercise, but a negotiation process where different viewpoints are brought to light and confronted with other perspectives. The ABS prototype was used to structure and organize the available information and simulation experiments, it was not possible to use this rudimentary suite to support collaborative planning. Furthermore it did not offer functionality to make annotations, share knowledge, or organize the work. In fact it mainly offered services for analysis and designing solutions, but it lacked services to evaluate and compare alternatives.

2.1.5 Conclusions

In this case study we explored the challenge of leveraging the expert knowledge and information to the executive level using a suite of simulation services: the ABS. The ABS was grounded on the idea that existing analytical- and simulation models should be integrated into a consistent suite of services. Based on our evaluation we drew a number of conclusions with regard to the needs of studio based mainport planning.

The ABS focused mainly on supporting the activity of analysis by providing visualizations of model outcomes. However, the policy analysis approach also emphasizes evaluating and comparing alternatives, and making a choice for a preferred alternative. These activities are also part of the dynamic model cycle, and in chapter one we argued that studio based decision making should put an equal emphasis on all stages and activities. During the evaluation of the ABS prototype it was confirmed that we should have put a stronger emphasis on all activities. We therefore concluded that there is a need for specific visualization services to support the activities of analysis, design of alternatives, evaluation, and choice.

The knowledge and information presented using the ABS has many different forms: demand for flights, flight schedules, capacity graphs, noise contours, economic tables and bar charts. Some information is more on an operational level (capacity and delay) while other information is more on a strategic level (economic figures). Furthermore there is static information (air traffic regulations) and dynamic information (flight schedules). We conclude that there is a need to visualize information from many different points of view, making use of appropriate visualization techniques. These techniques include graphs, charts, diagrams, maps, animations (e.g. of simulation outcomes) and interactive visualizations.

Further analysis in the ABS is focused on visualizing the model outcomes such as airside capacity and delay and economic profit, however, in a what-if analysis many different policies are analyzed under a variety of scenarios. The outcomes can not be interpreted without knowing the context: that is the scenario used, the considerations, assumptions and deliberations of the involved actors. Therefore we concluded that there is a need for functionality to support making annotations, comments and linking model outcomes to other sources of information such as documents, reports and memo's.

Finally, as more scenarios are analyzed, and more policies and alternatives are evaluated there is a need to structure the information. We designed a rudimentary structure in the ABS to organize the information in terms of simulation experiments. However, we found no evidence to show whether this was an appropriate structure. We concluded that we needed to investigate further how we can structure the information which is used and produced, and which accumulates, during the planning process.

2.2 Supporting area planning in the Port of Rotterdam

2.2.1 Introduction

The planning of land allocation in the Port of Rotterdam (PoR) is a complex task. There is a risk that the issuing of land when a potential customer arrives will cut the available resources to bits and decreases the synergy between companies in the port industrial zone. In this case study we focused specifically on a planning process called area planning. Area planning is aimed at developing a plan for land allocation for a specific area in the port. Usually area planning takes about 10 to 12 months and involves specialists, "area planners" who each provide input from their own disciple. Due to the relatively long throughput time it is difficult to keep all area planners involved for the entire period of the planning process. Furthermore there is no guarantee that knowledge gained during an area planning process will be available for future planning processes.

This case was characterized by the involvement of actors with backgrounds in specific disciplines such as infrastructure management, geovisualization, commerce, safety and environmental studies. Hence the available knowledge and information fell into many categories. During an area planning process knowledge and information accumulates leading to an information overload which clouds the completeness and punctuality of the available information. The PoR wants to disclose the richness of the available knowledge and information held in their organization; they want to consider the information in its integrality, not isolated in separate disciplines. Furthermore, they want to rehearse the future and possible transitions to the future by effectively analyzing and evaluating alternative designs for land use under a variety of scenarios.

The PoR actively supports the concept of studio based planning to realize these needs, which underlines the relevance of this case. We were not limited to the academic environment, but we could interview area planners and their managers, we could observe a real planning process being executed, we could access previous planning reports and information, we could get direct feedback on design ideas, and we were offered the time and resources to demonstrate our ideas in the form of a prototype.

Our challenge in this case was to improve our insight and understanding of mainport planning, and to show how such a planning process could be supported using a suite of services. We took what we had learned from the ABS case, and moved the focus of the case study to the studio as a whole. Instead of designing a suite to support the planning process, this case offered a rich environment in which to focus on the blend between people, processes and technology. Our perspective was from the inside-out, instead of outside-in to determine the domain experts involved, their way of working, the technology currently used by these experts, and what are their ideas on studio based planning were.

2.2.2 Area planning

We started with an investigation of area planning and planning in general in the PoR. From the very beginning of the case study we had regular meetings with a core response team in the port which was lead by the head of the Port Infrastructure division. The role of this team was firstly, to provide information on area planning, secondly to establish possibilities for interviews with key experts, observations and presentations, and thirdly at a later stage to provide direct feedback regarding our design ideas. To do this we studied already completed area plans and related documentation. Interviews were conducted with experts from different departments and divisions such as the design and drawing office, infrastructure management, environmental affairs and commercial affairs. Also, Schalkwijk (2004) studied and documented area planning meetings as part of his masters graduation project and thereby contributed to our investigation.

Types of port planning

The PoR carries out planning processes for land allocation at several levels (Smits et al. 2005) which can be characterized by the size of the the area considered, the level of detail of the allocation, and the period of time for which the plan is made. A port plan or master plan is a long term (20 year) high level vision for a complete port and describes how the port is related to its surroundings. A regional plan, or labeling plan is aimed at a specific region in a port and it describes the possibilities for land allocation at a high level, labeling areas for specific industry types. Area plans give more precise directions for further development of an existing port area for a 5 to 10 year period. The objective is to find a match, a carefully considered balance, between expected customer demand and the supply of land. Area plans typically describe the location of (future) customers, facilities and infrastructure investments. Finally, a development plan describes the steps that need to be taken towards the realization of goals and the planning of contracted work for a period of about 4 years. Such a plan usually takes an area plan as input.

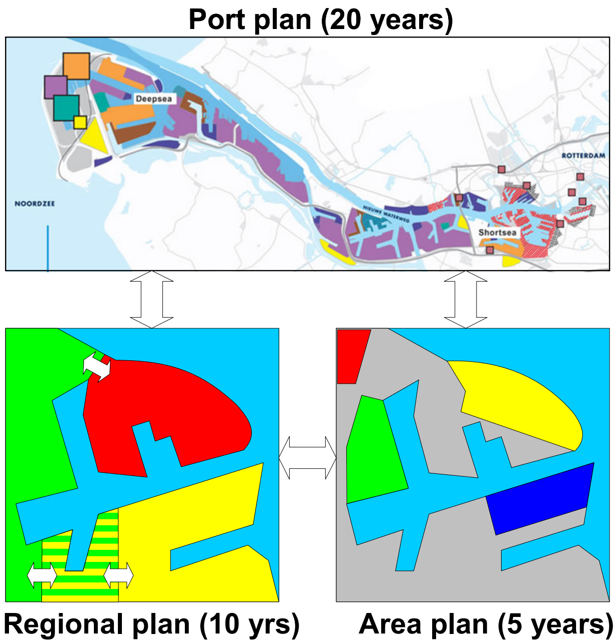


Figure 10: Different levels of port planning in the PoR (source: Chin et al. 2005c).

The actors involved in port planning

Area planning is a teamwork activity. In the Port of Rotterdam, an inter-disciplinary team of experts is set up, in which different roles can be distinguished such as project management, domain experts, analysts, information specialists and decision makers. A person can have more than one role at a time, and during the process the people involved and their roles may change. Different fields of expertise are represented (Figure 11) by domain experts in domains such as geographical information, commercial affairs, infrastructure, traffic (road, rail, water) and environment impact.

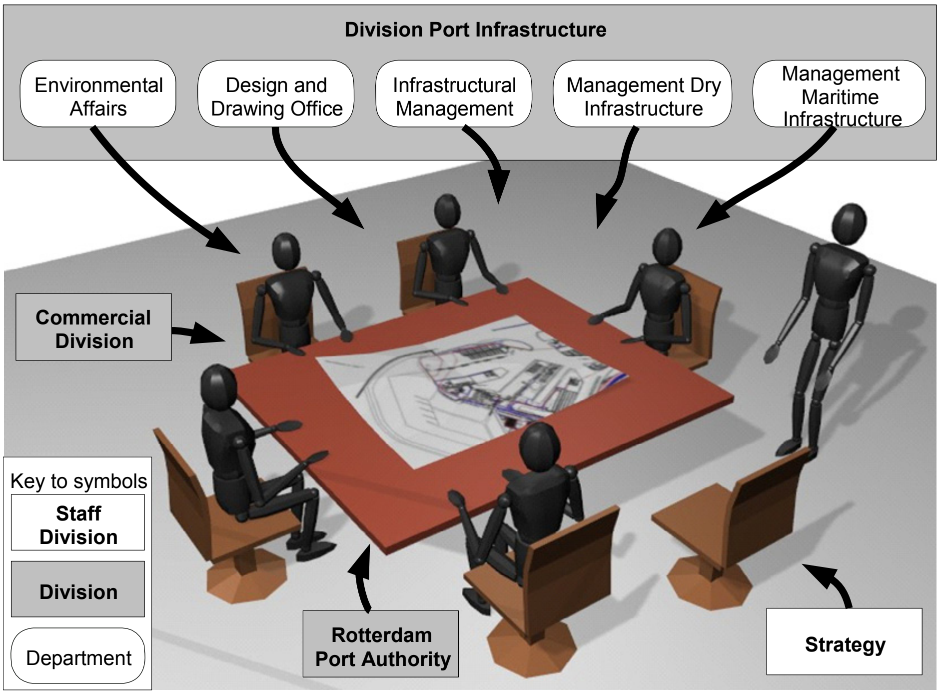


Figure 11: The actors involved in area planning (source: Chin et al. 2005c).

The process of area planning

Planning in the port is conducted in the form of projects. Different projects may run simultaneously and often partly overlap. For example, one team can be developing a port master plan while another team is making an area plan. Port planning is considered to be a long term process e.g., an area planning project can take up to ten months. The outcome of a project is a report describing a desired future situation, usually incorporating geographical maps. This report is used to provide information for management, and to guide future investments and commercial activities. Area planning can be subdivided into a number of stages. First a startup is made by identifying the problem area and setting up an initial project team. After this the area is investigated and analyzed to gain insight into the available resources and issues. Next the opportunities and constraints of the area are identified and worked out in a number of designs. Workshops are setup for key experts aimed at converging knowledge and information to produce an acceptable area plan. Finally the area plan is documented and presented to management and the commercial affairs department of the PoR.

Area planning is a distributed process. A team of area planners is formed consisting of members from infrastructure management, environmental affairs, maritime infrastructure, commercial affairs and so forth. They assemble in multi-disciplinary meetings to collaborate, negotiate, communicate, deliberate and argue about an area from many different viewpoints and with often differing objectives. They design by making design sketches, drawing contours and marking areas on geographical maps using their knowledge, expertise and common sense. Next they go their different ways, to work as individuals from behind their desks to do detailed analysis, investigations, make models and write documents which are then used as input for the next meeting. They assemble and distribute, working synchronously as a team and asynchronously as individuals.

Information used in area planning
Area planners both make use of existing information and produce new information. They use domain specific information available within the departments of the organization. The information can be in virtually any form including geographic maps, technical reports, development plans provided by the municipality and information obtained by accessing corporate databases (such as customer information). In conjunction information is produced in the form of design sketches, maps, memos, reports, models etc..
To find a match between customer demand and the supply of land, information regarding a wide variety of aspects needs to be considered (Figure 12). These aspects represent both hard quantitative information and soft qualitative information. Consequently possibilities are determined by physical constraints, and by the desirability to invest in overcoming these constraints. For example, investment in a noise damping wall may cause certain "noisy" companies to become viable candidates for a location in the port, where before this was not possible due to noise restrictions. Aspects such as the commercial attractiveness of a customer are mainly characterized by the strategy of the port organization and less by physical constraints or regulations.

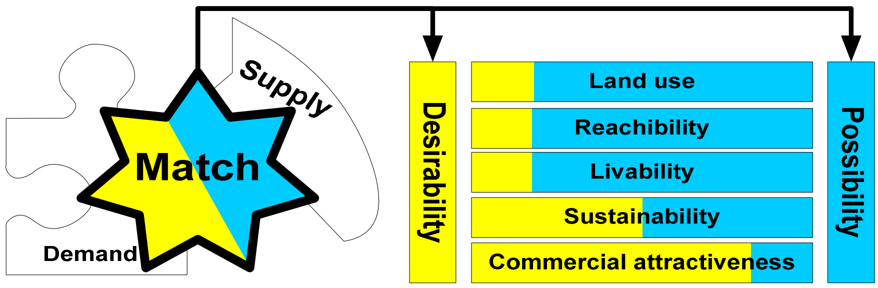


Figure 12: Aspects regarding land allocation in the port (source: Schalkwijk 2005).

The use of technology in area planning
Currently IT technology is little used during PoR area planning meetings. The information used in area planning is mainly paper-based, little is presented electronically. Area planners rely on geographical material, usually in the form of paper maps that are tailored to their specific needs. A map provides a focal shared viewpoint for all involved actors. At the Port of Rotterdam, the drawing department uses advanced GIS software to produce these maps, however these systems are not used directly during the planning process. Design sketches are made directly on paper maps. After an area planning meeting the design sketches are worked out by the drawing department and turned into official design drawings.

The challenge of supporting area planning
A number of challenging issues appeared or were confirmed during our analysis. Currently, a port planning process can take up to ten months. However, as the planning process progresses, the world keeps changing and management and commercial affairs need to make decisions in a shifting environment. In the end they are informed by a report, but this provides them with a static image of a future situation which may even at this time be outdated due to new developments. Therefore the challenge is to decrease the time needed for port planning and to find ways to support dynamic updating of port plans as the situation changes. Ideally port plans should be updated continuously with reliable data, providing insight into the future situation and into how the transition to the future will be made.

Another challenge is to increase the ability of managers and planners to learn from previous projects, which would help to reduce the time required to produce plans and help to structure the planning process. Currently only the outcome of a planning process is documented, but not the process: information on the intermediate steps which led to the end result, and the alternative solutions which were rejected earlier in the process is not kept. An abundance of information is used and produced, however, not all of it is organized in such a way that the actors involved can easily find and retrieve relevant information. This makes it difficult to trace back how certain conclusions came about.
Furthermore it is a challenge to provide an integral view of the different aspects (e.g. accessibility, economics, sustainability etc.). Currently the different aspects are analyzed mostly in isolation. As a result, it is for example difficult to determine the effects of increased road traffic on air pollution and how it relates to other causes of air pollution.

2.2.3 Design

During early discussions at the PoR the core response team explained their ideas on supporting area planning using a suite of software services. A rather sculptural description was provided by the division head. He told us that ideally he would like to be able to lift up a company in the port from a 3D computer visualization and see all the wiring, pipelines, communication lines, and traffic connected to it. Furthermore it turned out that for a specific area planning project regarding an area called the NW Hoek, the project leader had developed an Excel spreadsheet which was internally called the “Matrix”. The Matrix is a table where the rows represent possible industry types and other types of land allocation, and the columns represent the relevant aspects that should be evaluated. The cells of the Matrix are filled using expressions representing common properties of combinations of industry types and aspects. For example, a cell representing a container terminal and lot size contains the expression 80-140 Ha, which indicates the expected surface required. The Matrix has been found to be very useful for making a quick scan for possible industry types on a certain lot. Furthermore it is used as a checklist to make sure that all relevant combinations have been addressed. Area planners are used to taking a paper print of the Matrix to area planning meetings.

These two concepts, the “3D computer visualization” and the Matrix, combined with the lessons learned from our analysis of the process and the concept of studio based decision support (chapter 1) formed the staring point for a first design of a suite of services which was initially called a Studio for Area Planning (Chin et al. 2005c).

We regard the suite of services to be a distributed environment, a virtual space which links people and information using the organization’s intranet (Figure 13).

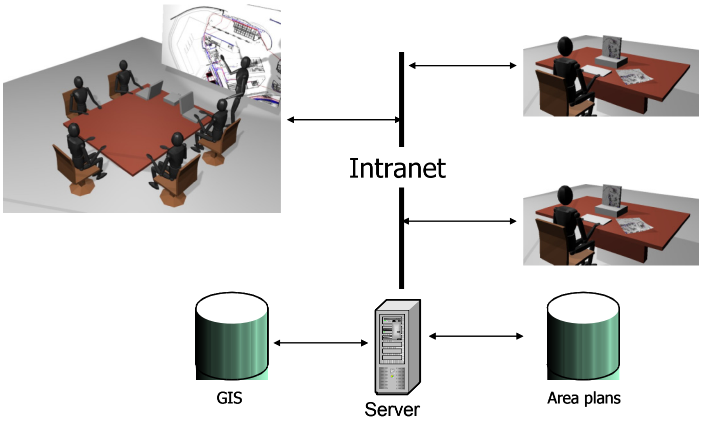


Figure 13: The studio as a distributed environment.

An early architecture concept for the suite of services (Figure 14) contained services such as:

- infrastructure manager: to configure the infrastructure in the area
- scenario manager: to select which situation is analyzed
- version manager: to enable multiple users to access and update information
- data manager: to help the connection to available data sources in the organization
- aspect model adapter and aspect model chooser: to provide users with access to a library of models

These services were linked through a backbone to data linkage services which were used to provide access to various data sources and GIS services.

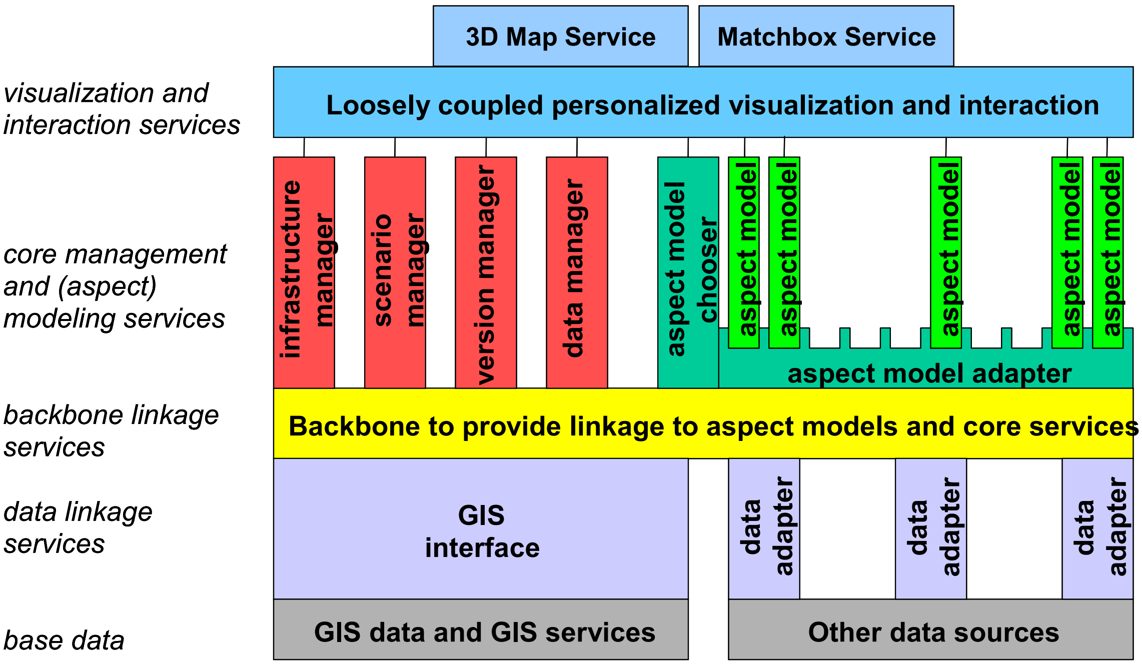


Figure 14: An early architecture of a suite of services to support area planning

These components were provided using a loosely coupled, web-based visualization and interaction layer, to provide two important services:

- a digital 3D geographical map of the area, companies, and infrastructure resources
- a fully configurable, interactive and digital visualization of the Matrix which we call Matchbox

In contrast to the use of paper maps the digital geographical visualization enabled users to interactively aggregate information from various types and sources. It enabled users to layer information and literally see the interrelatedness of e.g., traffic and NOx levels. Furthermore changes to infrastructure and companies in the area could be immediately be visualized, providing direct insight.

Matchbox is a visualization service which adds interactivity to the original concept of the MS Excel based Matrix. While the Matrix is a static checklist showing the key characteristics of a fixed set of industry types, Matchbox is fully configurable for a specific location. Alternative industry types (solutions) and the relevant aspects can be fully configured.

Not only can the Matchbox be configured, it can also provide insight into the status of an evaluation (Figure 15). The evaluation of alternative solutions is supported in two iterative stages. First individual domain experts make evaluations of the alternative solutions for an aspect in their own field of knowledge. A domain expert makes the evaluation by matching the specified aspect with the available alternatives. He or she uses a color to encode the outcome of the evaluation: red, yellow, or green. Red means impossible or undesirable, while green means possible or desirable. Yellow is used to indicate that no definitive outcome could (yet) be defined. Each evaluation of an alternative-aspect combination can be the outcome of an extensive analysis, which is based on the available information and knowledge. Therefore it is important to record the outcome and the thinking behind an evaluation. This is achieved by providing the experts with a facility to drill down into the cells of the Matchbox and textually motivate each evaluation.
The second stage of evaluation is a multi-disciplinary stage. A team of experts and managers joins in a meeting where they aim to differentiate between those alternatives which are viable and those which should be rejected. In this process of Singerian inquiry the alternatives are queried from many different viewpoints. The outcome of this deliberation is visually encoded by dragging the alternatives to either the “approved category”, or the “rejected category”. Finally, no alternatives should remain in the category “under evaluation”. During a multi-disciplinary meeting it may be found that some alternative-aspect combinations should be re-evaluated, or new alternatives or aspects may be introduced for consideration. This leads to an iterative process which alternates between individual expert evaluations and multi-disciplinary team evaluations.

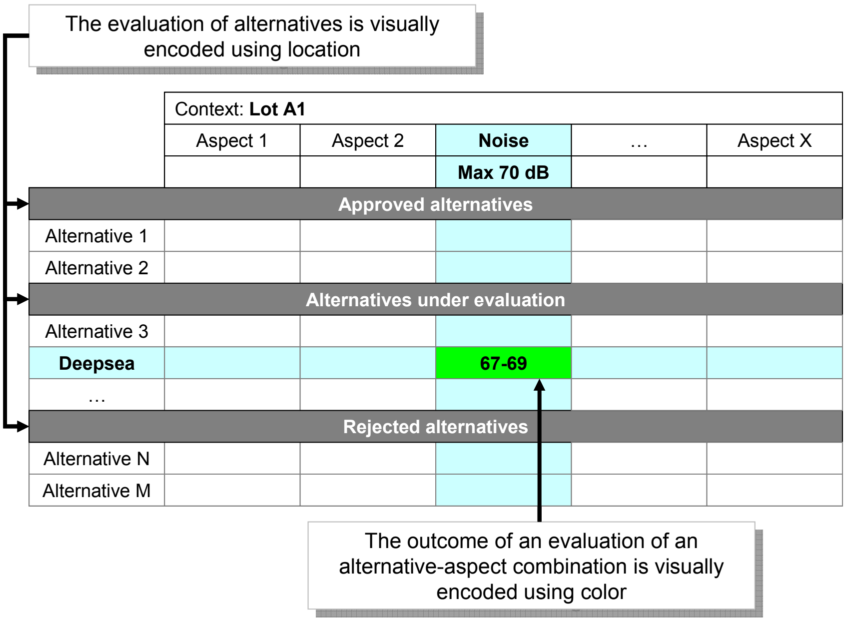


Figure 15: Matchbox, a visualization service for evaluating alternative solutions.

2.2.4 Evaluation

We implemented a rudimentary prototype of the suite which we used for demonstration and gain feedback on our initial design ideas. The prototype was a realization of the visualization and interaction layer. The underlying services were worked out in as far as is required to demonstrate the functionality of the suite. The prototype demonstrated the use of the digital map and Matchbox service in a web-based suite. In this section we evaluate our ideas on studio based area planning.

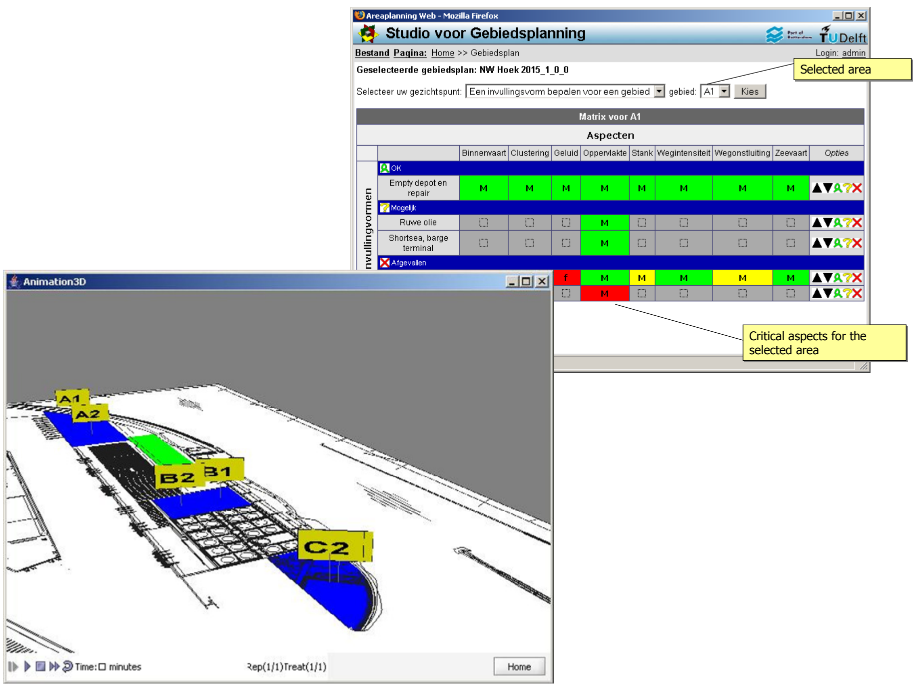


Figure 16: The prototype of a suite of services to support area planning.

We proposed a web-based area planning suite because we found that both the people and the information involved are distributed. The people work synchronously as a multi-disciplinary

team and asynchronously at their individual office desks. They use information drawn from throughout the organization and produce design sketches, reports, memo’s and other information. Some remarks of the interviewed area planners were: that there is a need for “digital access to quickly disseminate information”, and that “it would be nice to take your laptop with you so you can quickly show available lots and the most important characteristics of these lots”. Furthermore, the drawing department confirmed our decision for a web-based solution. They are responsible for updating and providing geographic material in the port. However, design sketches hand-painted on printed maps tend to end up in desk drawers and to get lost, or take on a life of their own. Experts from the design and drawing department told us that a web-based solution would reduce the possibility of loosing track of geographic information allowing them to record design ideas directly in a central project database which can be accessed via the organization’s intranet.
Furthermore, the PoR already makes extensive use of web-based technologies. The drawing department used RIV, a system which can be used to visualize detailed location based information on the current situation in the port. The commercial department manages a customer database which is accessible through the intranet, and the dredging department uses the intranet to keep track of water depths. However, there is not yet a specific set of services which brings together all relevant resources in a convenient way for area planning.

We demonstrated the prototype during a meeting with the commercial division and the port infrastructure division. During this meeting the participants stressed the value of visualizing the developments in the port area, both for internal planning purposes and as a means to provide information for potential customers. What needs to be visualized is the interrelatedness of the information: e.g. the synergy between companies in the port industrial zone, and the dynamic and transitional changes in the area especially regarding traffic density. Furthermore the PoR area planners want situational awareness. They need to answer questions such as: What is the status of this map? What information is related to this scenario? How old are these safety contours? What the area planners do not want to see is a system which forces them to focus on issues which are irrelevant, or that limits their creativity with bureaucratic distractions. For example, they do not want to fill in noise levels for an area which is located far away from urban regions, and they would not want to evaluate the option of a large container terminal for a small city harbor. Furthermore, we noticed a difference between the needs of the commercial division and the port infrastructure division. The commercial division takes a customer viewpoint and asks: If we got this commercially attractive customer then which would be an appropriate location? The port infrastructure division takes the opposite viewpoint and asks: which type of customer would ideally be located at a specified lot. One of the interviewed area planners stated: “The challenge is to support the many different viewpoints such that the involved actors can converge to a balanced solution.”

Finally, an often returning issue is information overload, or a lack of situational awareness. The area planners indicated that they do not always know where information is stored, how old the information is, in which context it was produced, who owns the information, or even if the information is available. In addition information of many kinds is produced during the planning process. The information relates to content i.e. maps, reports, annotations etc. and to the planning process, i.e. its documents: what was agreed upon, who does what, the report version, what was discuss last meeting, what issues are still open etc.. Currently there is no consistent record of area planning project information, but as area planning becomes more important in the organization there is a need for a more streamlined information flow. As Oosterhuis (2003) states: “You must run the process and work in the process”.

2.2.5 Conclusions

In this case we aimed to gain a better insight into and understanding of studio based mainport planning. In line with the previous case, the ABS, we saw that mainport planning involves analysis, design of alternative solutions, evaluation and then choosing a preferred solution which is presented in the form of a report, a plan. We saw that there is a need to visualize information from many different points of view. Annotating and documenting maps, sketches and design ideas is considered important. Furthermore port planning involves a large amount of information. In addition, the rich setting of the case in the PoR gave us some new insights which we discuss below.

In area planning people work as a multi-disciplinary team. However they do not always work together in the same room. They prepare, analyze, and evaluate information in their own discipline as individuals. In regular meetings they share, explain, discuss and evaluate their individual findings as a group. They work synchronously and asynchronously and they use information which is scattered all over the organization's databases and archives. Therefore we concluded that there is a need for a web-based solution which can be accessed through the organization's intranet that lists relevant information sources.

The area planning team members play different roles in the planning process. Most team members take the role of domain expert in their own discipline, while other assume the role of project leader, meeting facilitator, analyst, modeler or administrator. During the planning process people may change roles, people leave the project and people are introduced into the project. Furthermore we saw that the involved actors were interested in different types of information in different area planning projects. We conclude that the suite should support the different roles and changing information needs of the actors involved in mainport planning.

In the ABS case we concluded that we needed to provide the possibility for actors to keep track electronically of annotations, remarks and documentation related to the outcomes of an analysis. In the PoR case we saw that a planning project stretches over months. Area planners have to agree on actions and decisions regularly. We therefore concluded that there is a need to document the outcomes of an analysis and of the planning project itself: what are the actions to be taken for the next meeting, what was agreed upon, which issues are still open?

Finally we saw that area planners rely on geographic maps. There is a paper map of the port area on the table at almost every meeting. Area planners draw on these maps to exchange and communicate design ideas. Specific maps are prepared by the drawing department to provide insight and understanding in specific issues such as safety, pollution and water levels. Furthermore each meeting room has a map of the PoR on the wall. The map acts as a shared reference point, something all experts can identify with no matter their field of expertise. The map is a writing-pad for domain specific layers of location based information. We concluded that there is a need to provide and electronic geographical map as a shared reference point and writing-pad for all the actors involved.

2.3 Summary and revision of research question

In this chapter we investigated the needs for leveraging expert knowledge and information to the executive level in studio based mainport planning. We executed two case studies to investigate mainport planning in practice (Chin et al. 2005a). From our investigation we gained a better insight into and understanding of mainport planning, which we used to extract the needs for a suite to support studio based mainport planning, these are listed below. There is a need:

- for visualization services which support the planning activities of analysis, design of alternatives, evaluation and choice
- for services to support making annotations, comments, and keeping track of actions, decisions and progress
- to structure, organizing and archiving the accumulation of information used and produced in mainport planning,
- for a web-based solution as people and information are distributed over the organization,
- to customize the way in which information is presented based on the actor's role and the relevant issues in a mainport planning project
- to visualize information from many different points of view, using different visualization techniques such as graphs, charts, diagrams, maps but also animation and interactive visualizations
- for a geographical map as a shared viewpoint and writing-pad for all involved actors

In chapter one we introduced our initial research question. Based on our case study findings we can now revise our initial research question into a number of sub-questions.

Visualization of different types of information for actors in different roles is a key characteristic of the identified needs. We did not yet address the theoretic underpinnings of visualization. Our first objective is now to search for principles in the field of visualization which we can follow for the design of our suite. In this chapter we described two exploratory case studies which represent two design cycles. Hence we do not enter our search completely blank, instead we can do a directed search for relevant principles in the field of visualization. This leads us to our first research question.

Research question 1:
What principles in the field of visualization can we use in the design of a suite to support mainport planning?

The principles in the field of visualization which we can follow to design our suite eventually need to be implemented using IT technologies. From our exploratory case studies we can conclude that the needs of mainport planners are not directly targeted at specific visualization technologies. We need to make the translation between what is needed by the mainport planners and what is available to support their needs. This leads us to our second research question.

Research question 2:
What are promising technologies which we can use to implement the identified principles in the field of visualization?

Designing and implementing a suite to support studio based mainport planning is a necessary step to be able to answer our main research question. Considering the different needs of actors involved in mainport planning and the diversity of planning projects, we argued that our design should be based on loosely coupled services for visualization and interaction. This leads us to our third research question.

Research question 3:
Can we design and implement a suite to support studio-based mainport planning using an architecture which is based on loosely coupled services for visualization and interaction?

Finally we need to come back to our main question and provide evidence that our suite improves the effectiveness of the actors in mainport planning.

Research question 4:
Can we provide evidence that such a suite improves the effectiveness of the actors in studio-based mainport planning?

In the following chapters we subsequently address these research questions, starting with an investigation of the field of visualization in the next chapter.

3 Visualization

In this chapter we address the first research question:

What principles in the field of visualization can we use in the design of a suite to support mainport planning?

To answer this question we took what we learned from our case studies in chapter 2, and searched for literature in the field of visualization. Our goal was to find those visualization principles that we can use for the design of our suite. In chapter 2 we concluded that we need to visualize different types of information for actors in different roles. Hence, we searched for those visualization principles that can fulfill this need, and that can serve as a basis for the design of our suite. We first focus on defining visualization in 3.1. Next we describe the field of visualization in 3.2. In 3.3 to 3.6 we address the underlying principles, techniques and best practices used in visualization. Finally we summarize our findings in 3.7.

3.1 Visualization and visualizations

Tufte (1990) states that while the world is complex, dynamic and multi-dimensional, paper is a static and flat medium. He starts his argumentation for an impressive categorization of visualization techniques, by wondering how we are to represent the rich visual world of experience and measurement in what he calls mere flatland? In posing this question Tufte addresses the very essence of visualization. Visualization is an artifact, a tool used to depict some of the complexity of the real world.

Keller and Tergan (2005) mention the role of visualization in helping people in processing, allowing them to gain access to, and dealing effectively with complex knowledge and large amounts of information. Often the well known phrase “a picture is worth a thousand words” is used to explain the value of visualization. Spence (2001) explains visualization as a cognitive activity, an activity that goes on in the human mind: the forming of a mental image rather than of one on paper. He states the potential of visualization as that of gaining insight and understanding. Visualization and cognition go hand in hand as is explained in the work of Strothotte and Strothotte (1997). Cognition refers to the question of how humans recognize information and associate situations that are recognized from their past experience. Card et al. (1999) explain cognition as: “the acquisition or use of knowledge”. Cognitive science, however still lacks a definitive answer as to how cognition actually works. Strothotte and Strothotte (1997) advocate the communicative framework of Weidemann, which stresses the communicative aspects of graphical representations. More recently, Ware (2005) has introduced the theory of visual queries as the foundation of visual thinking. He argues that although humans have the compelling illusion of being aware of the complexity of the world, humans can actually only keep about three objects in their working memories from one second to the next. He states that: “the impression that we have of a detailed visual environment comes from our ability to make rapid eye movements and sample the environment at will”. Depending of the task at hand our brain formulates a visual query on the environment. We see what we want to see by searching for patterns in the environment. According to Ware taking visual queries as a central concept for human perception will change the way in which we think visually with interactive displays.

Keller and Tergan (2005) explain that the term (information) visualization has different meanings depending on the context. They mention that psychologists use the term to signify a representational mode, referring to visualization as a cognitive activity. We followed Card et al. (1999). They define visualization in what Keller and Tergan call the "narrow sense" as:

"The use of computer-supported, interactive, visual representations of data to amplify cognition."

Card et al. (1999) argue that, supported by a number of examples, visual artifacts have profound effects on peoples' abilities to assimilate information, to compute with it, to understand it and to create new knowledge. Furthermore they state that: "Visual artifacts and computers do for the mind what cars do for the feet or steam shovels do for the hands. But it remains to puzzle out through cycles of system building and analysis how to build the next generation of such artifacts".

3.2 Fields of visualization

Card et al. (1999) provide an extensive history of visualization. They mention that in 1985 the National Science Foundation (US) launched a research initiative on scientific visualization. McCormick et al. (1987) were the first to advocate this federally funded initiative and coined the term visualization. They describe a number of research opportunities where they made a distinction between scientific opportunities and engineering opportunities. The scientific opportunities concerned fields such as molecular modeling, medical imaging, mathematics and geosciences. The engineering opportunities concerned fields such as computational fluid dynamics and finite element analysis. They mention topics such as interactive 3D images in molecular modeling and medical imaging, and mathematics.

A number of subfields of visualization emerged. The original field of visualization is mainly based on the visualization of physical data, such as weather conditions and measurements on the human body. This field is now referred to as **scientific visualization**. Adjacent and partly overlapping to scientific visualization is the, now predominant, field of **information visualization**. Information visualization can be distinguished from scientific visualization because it addresses non-physical information such as financial data, business information, collections of documents, and abstract concepts (Card et al. 1999). Spence (2001) argues that information, or the information explosion addressed in information visualization, is actually a data explosion: "it is the derivation of information (or understanding or insight) from the data that is difficult, and which we attempt to facilitate by means of visualization tools". Keller and Tergan (2005) state that information visualization is an umbrella for all kinds of visualizations regarding the processing, comprehension, and retention of information in static, animated, dynamic and interactive graphics. They stress the increasingly important need to have information 'at your fingertips': "There is a need for cognitive tools aiming at supporting cognitive processing in generating, representing, structuring and restructuring, retrieving, sharing, and using knowledge. Therefore there is a need for visualization techniques for making structures of information in large repositories apparent and for helping users in effectively searching and locating task relevant information elements while coping with large amounts of information in learning and problem solving."

More recently a new subfield of visualization appeared called **knowledge visualization** (Burkhard 2004a). Burkhard explains that knowledge visualization focuses on the "transfer of knowledge to different stakeholders; especially the transfer of insights derived from information visualization tools". He specifically mentions the potential of knowledge visualization in architecture, where he argues that the transfer of knowledge between planners and decision makers can still be improved (Burkhard 2004b). Furthermore he defines a knowledge visualization framework "to systemize research, to identify research gaps, and to mediate between researchers from different fields". The elements of this framework constitute:

- **visualization objective**: why should knowledge be visualized e.g. for coordination, attracting attention, supporting recall, to motivate people, elaboration, or creating new insights
- **knowledge type**: what content needs to be visualized, or as Burkhard puts it: know what, know where, know why, know how, and know who
- **visualization type**: what is an appropriate type of visualization to represent a certain type of knowledge e.g. sketches, diagrams, images, maps, or (visual) stories
- **recipient**: for who was the visualization created e.g. individuals, groups, organizations, or networks of people

Canas (2005) explains the interpretation of the term 'knowledge' as concepts (Margolis and Laurence 2003) and the relation between concepts, called propositions. Propositions form a linking relationship between concepts to form a semantic unit. Directly related to this interpretation of knowledge is the concept map, or semantic map, which is a spatial representation of concepts and their interrelationships intended to represent the knowledge structures in human minds (Jonassen 2005). For example, Novak and Wurst (2005) make use of concept maps to support cross community learning, and Canas et al. (2005) make use of concept maps to browse and organize information repositories.

Apart from the described subfields we can also distinguish a number of fields which are closely related to, and probably partly overlap with the field of visualization. For example, we consider the field of **geographic visualization** and thematic cartography to be relevant for this research, because of the reliance on cartographic material in mainport planning (chapter 2). Although the name geographic visualization suggests that it is actually a subfield of visualization, Slocum et al. (2005) argue that the origins of the field date back much further. They mention that the ideas now popularized in information visualization were already addressed by cartographers in the 1950s. Furthermore visualization is closely related to the field of Human Computer Interaction (HCI) which we will address in section 3.5, and finally the field of Computer Graphics (GC) (Foley and Van Dam 1982) can be considered as the provider of the enabling technology, which eventually will result in more advanced visualizations.

3.3 Visualization principles

Jaesckle et al. (2005) argue that information visualization and knowledge visualization employ comparable techniques and methods. Furthermore studying the work of Slocum et al. (2005), we can conclude that the field of geographic visualization and thematic cartography is based on similar techniques and methods. Jaesckle et al. (2005) explain that over the years a variety of classification schemes were proposed to increase the understanding of the principles underlying information visualization. They provide an extensive overview of these several classification models and the interrelationships between these models, which they present in a diagram. This diagram (Figure 17) shows that there are many information visualization models with often partly overlapping characteristics. We argue that it is not possible nor relevant to provide a complete overview all currently existing classification models within the context of this thesis, instead we will only highlight some of the most elementary models of visualization.

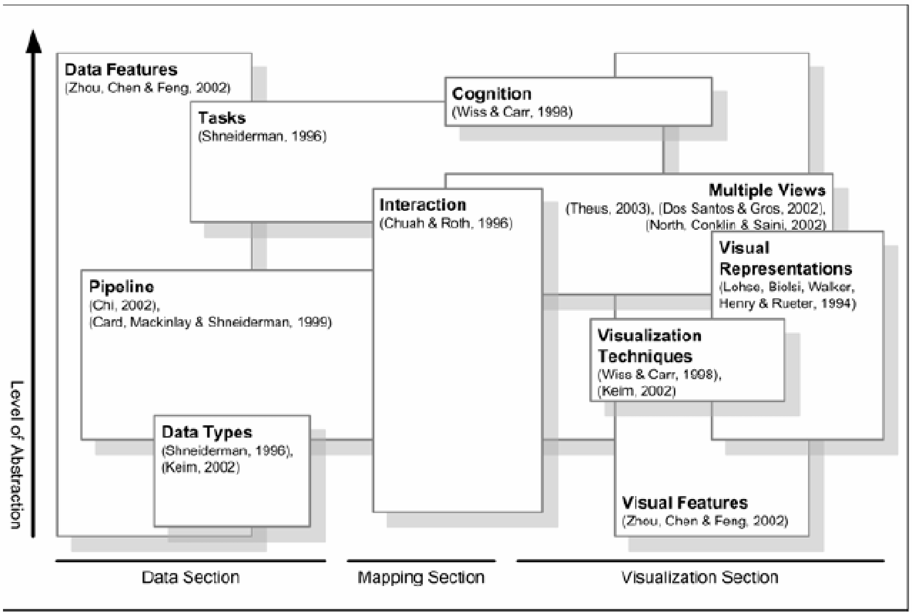


Figure 17: Interrelationship of information visualization models in information visualization model space (Jaeschke et al. 2005).

A basic reference model for the process of visualization was introduced by Card et al. (1999) (Figure 18). This model provides a high level view on the process of information visualization. A mapping from data to visual form is performed in a number of steps. First raw data is structured, filtered and possibly normalized. The resulting transformed data is mapped onto a set of visual structures. Next the visual structures are used to generate a set of views which allow the user to navigate through the display, i.e. by specifying graphical parameters such as position, scaling, and clipping. User interaction can be of influence on all the steps mentioned before. For example, user interaction can restrict the view to certain data ranges.

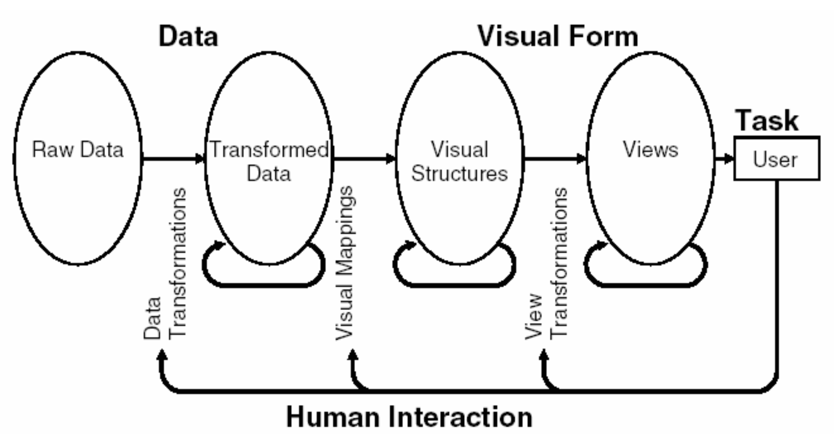


Figure 18: Reference model for visualization (Card et al. 1999).

Data underpins all visualizations as is explained by the basic reference model. Shneiderman (1996) made a distinction between seven types of data: 1-dimensional, 2-dimensional, 3-dimentsional, temporal, multi-dimensional, tree and network. Jaesckle deployed this taxonomy to sort out research prototypes for new information visualization design opportunities. Slocum et al. (2005) use an alternative categorization of data types which can be represented by visual elements: nominal (city names, categories), ordinal (days of the week), quantitative (real numbers) and relational. The relational type is equivalent to the tree and network types identified by Shneiderman.

The data which is transformed from raw data is visually encoded on a set of visual structures. Slocum et al. (2005) explain that underlying the creation of cartographic maps are the principles of symbolization, the proper use of graphical symbols to represent the types of data to model (geographical) phenomena. The visual variables, which can be varied, are spacing, size, perspective height, orientation and shape, arrangement, and color (hue, lightness and saturation). Furthermore a distinction can be made between 'retinal' properties such as spatial position, size, shape, color, orientation and texture; 'gestalt' properties such as connectivity and grouping; and animation such as transitions and animated elements (Bertin 1983, Card et al. 1999).

Next, the principles of symbolization are used to build visual representations, or views such as graphs, tables, time charts, network charts, diagrams, maps, icons and photo realistic pictures. With regard to geographic visualization three types of maps can be distinguished (Slocum et al. 2005).

- Choropleth: visualizes data collected for enumeration units such as countries
- Proportional symbol: visualizes data by proportionally scaling symbols to the magnitude of data occurring at specific locations
- Isopleth: visualizes data by using contour plots
- Dot: visualizes data by setting dots at places where a phenomenon is most likely to occur

In many cases visual representations are not static but allow the viewer to interact. Shneiderman (1996) enumerated seven interactive tasks which users can execute on visualized data.

- Overview: get an overview of a collection
- Zoom: zooming in on items of interest
- Filter: remove uninteresting items
- Details on demand: select items and get details
- Relate: view the relation between items
- History: keep a history of actions to support undo, replay, or refinement
- Extract: make sub collections

These tasks can be combined to define more complicated tasks.

3.4 Animation principles

In mainport planning time is an important dimension to be considered, but thus far we focused on static visualization principles. We now elaborate on dynamic visualizations, i.e. animation principles. Today animations are used for a wide variety of applications such as movies and computer simulations. Sturman (1998) states that as hardware becomes faster and less expensive and animation tools are improved, radically new ways of "doing animation" may be found. However, animation is not yet extensively addressed in the information visualization literature. For example, the readings provided by Card et al. and the more recent readings provided by Keller and Tergan do not provide a section on animation. Spence (2001) mentions animation several times as a feature of visualization techniques, but he does not address animation as a separate topic. In information visualization animation is often simply described as a type of visual encoding, putting animation on the same level as 'retinal' properties such as size shape and color. Yet visualization techniques such as the perspective wall (Mackinlay et al. 1991) and cone trees (Robertson et al. 1991) do rely on animation. Furthermore Slocum et al. (2005) extensively address the animation of geographical maps. Animation is a commonly accepted visualization technique in computer simulations (Schulze et al. 2005). Furthermore, commercial simulation software such as Arena™ and eM Plant™ provide animation functionalities.

Although not yet extensively addressed in information visualization, an extensive overview on computer animation provided by Pocock and Rosebush (2001) shows that the process of animation follows the same lines as the process of visualization as presented by Card et al. (1999).

Just as in static visualizations, animations are based on the underlying data; because animation has its origins in classic dramatic media such as film and television, the vocabulary of animation terms is largely shaped by these fields. The data represented in animations is organized in **frames**: sequences of images which are shown sequentially. A **shot** is a continuous sequence of frames showing an **action**, which is a change over time. Traditionally the carrier of animated data is called a **track**: an independent 1D carrier of information along the time dimension. Different tracks can be combined in parallel to form **scenes**, which in turn can be combined sequentially to form the different **acts** of a **play** (the complete animation).

The data are mapped on visual structures which may be the subject of an action. Slocum et al. (2005) distinguish between several types of animation:

- animations emphasizing change, e.g., the spreading of diseases through a certain area
- animations emphasizing location, e.g., the use of flashing points to locate the epicenter of an earthquake
- animations emphasizing an attribute, e.g., showing a map in sequence, piece by piece to show a certain order

Actually, only animations emphasizing change require time as a variable in the original data to visualize action. The other two types of animation show action, but these changes do not actually occur in the data filtered from raw data sets. Hence animation can be used to visualize dynamic data, or for visual encoding. Chang and Ungar (1993) go beyond the animation of data by utilizing animation to improve the usability of user interfaces. Pockock and Rosebush give two basic techniques that can be used to achieve action.

- Motion pathways, or motion curves: a line or curve along which an object moves through a sequence of frames
- Key frame animation: an animation method where action is bounded by a pair of extreme positions called key frames, the frames in between the key frames –the in-betweens- are extracted by means of interpolation (Burtnyk and Wein 1975)

The concept of easing is important in both techniques, this is used to describe how the in betweens are calculated. When an action starts slowly it is said to ease in, similarly when it ends slowly it is said to ease out.

Animated scenes often provide an abundance of information to the viewer. In traditional animations (films) a number of principles have been developed to help the viewer to understand complex scenes. Lasseter (1987) provides an overview of traditional animation principles and how these can be applied in computer animations, for example see below.

- Squash and stretch: emphasizing the rigidity in the objects when an object is moved
- Timing: showing the weight of an object which can contribute to the feeling of size, scale, or character of an object
- Anticipation: preparing an action to direct the attention of the audience to the 'right' part of the screen at the 'right' moment
- Staging: presenting an idea so that it is unmistakably clear, the audience eye needs to be led exactly where it needs to be at the right moment
- Follow through and overlapping motion: carrying an action past the termination point, as actions very rarely come to a sudden stop
- Slow in and slow out: visualizing second and third order continuity of motion
- Exaggeration: going to the heard of an idea and develop its essence
- Secondary action: showing an action which results from another action

Several studies on interactive simulations exist, where users directly manipulated simulation models through animation displays. Proof Animation is an early example of general purpose animation software meant to visualize simulation outcomes (Henriksen and Earle 1992), but this does not allow for interaction during a simulation. O' Keefe (1987) introduced Visual Interactive Simulation (VIS), which allows users to visually interact with a running simulation. Typically VIS provides visual output, user interaction and visual input. Predefined interactions enable the user to make changes to simulation models while running. In some cases it is even possible to change program scripts at run time. Graphical input refers to the ability to change graphically, or visually, program simulation models. Furthermore Vujosevic (Vujosevic 1990) includes an intelligent advisory function as part of VIS, i.e. capabilities to provide the user with guidelines on how to accomplish modeling and simulation activities. Several applications of VIS can be recognized: to argument selling by providing VIS as a presentation and communication medium, computer aided design, gaming with relatively complex system models, and enhancing learning by allowing users to play with the system (Bishop and Balci 1990).

3.5 Visualization and its relation to Human Computer Interaction

Spence (2001) states that all the visualization tools he discusses in his book had to be invented or designed as they were not generated automatically. However, he also recognizes a lack of understanding of Human Computer Interaction (HCI) and mentions that, in the great majority of situations, the design of new visualization tools is a craft activity, one that is highly dependent on the skills of the designer. Therefore it is no surprise that, for example, the works of Shneiderman can be found in publications in both the field of information visualization and HCI (e.g., Shneiderman 1998, Carrol 2001).
Visualization tools and techniques will eventually end up in user interfaces. Myers et al. state that modern user interfaces have their roots in the research of the 1970, 1980s and 1990s (in Carrol 2001). Today's window managers and graphical user interfaces found in e.g., MS Windows®, Mac OSX® and Linux, come from the research at the Stanford Research Institute, Xerox Palo Alto Research Center (PARC), and MIT in the 1970s. Myers et al. argue that the inventions of those years shaped the current "GUI style", and contributed to the development of event languages, interactive graphical tools, component systems, scripting languages, object oriented programming and even the use of hypertext. Shneiderman (1998) provides an extensive description of user interface design and related theories, principles, guidelines and best practices. Furthermore he provides methods that can be used for the evaluation of user interfaces. In addition Peterson (1995) explains the relation between geographic visualization, animation and user interaction. He argues that: "The map is no longer the static element in the communication chain. The map is being transformed into a more interactive, user-controlled type of display." This is why we had to recognize and address the relation between visualization and HCI.

3.6 Summary

In this chapter we addressed our first research question:

What principles in the field of visualization can we use in the design of a suite to support mainport planning?

We defined visualization in line with Card et al. (1999) as the use of computer-supported, interactive, visual representations of data to amplify cognition. We investigated the visualization field and its related subfields: scientific visualization, information visualization and knowledge visualization. We described best practices, underlying principles and techniques in relation to visualization. Furthermore, we investigated the possibilities of geovisualization and animation. Finally we briefly addressed the fields of HCI and its relation to visualization.

The principles and techniques presented in this chapter form the basis for the design of the mainport planning suite. The reference model of Card et al. and the klowledge visualization framework of Burkhard were used to pinpoint the requirements for a mainport planning suite in the following chapter. In the next chapter we describe how the principles of visualization can be mapped to requirements for a suite to support studio-based mainport planning.

4 Requirements for a Mainport Planning Suite

In the previous chapter we investigated the principles of visualization. We specifically focused on information visualization, knowledge visualization and geovisualization. In this chapter we will use the identified visualization principles and the needs which we identified in chapter 2 to extract the requirements for our suite to support mainport planning. In doing so we address our second research question:

What are promising technologies which we can use to implement the identified principles in the field of visualization?

We follow the lines set out in the visualization reference model of Card et al. (1999) and the knowledge visualization framework of Burkhard (2004a,b). While Card et al. focus on visualization as a process, Burkhard focuses on the exchange of knowledge and information through visualization. Combining both approaches allowed us to create a framework from which we can extract the requirements for our suite (Figure 19).

We start in section 4.1 with defining our system boundaries: Which activities do we choose to support with our suite and which ones do we leave out? Next we describe how we will access the information and knowledge needed in a mainport planning project. In 4.2 we describe how raw data can be identified and indexed. In 4.3 we describe how we have structured, filtered and transformed the raw data. As we will explain in 4.4 meta-data plays an important role herein. Therefore we also refer to the identified need to make annotations. The next step was to map visual structures onto the data to produce different visualization types, which we will describe in 4.5. The final step in Card et al's reference framework is to create views from the visual structures. Burkhard's framework stresses the role of the recipient, which is why we address the creation of views for distributed actors in different roles in 4.6 and 4.7. Finally we extract the requirements for our suite in 4.8.

		Information visualization				
			Raw data	Transformed data	Visual structures	Views
Knowledge Visualization	Visualization objective	4.1				
	Knowledge type		4.2	4.3, 4.4		
	Visualization type				4.5	
	Recipient					4.6, 4.7

Figure 19: Combining Card et al's visualization reference model and Burkhard's knowledge visualization framework to identify the requirements for our suite, and the section numbers where each combination is discussed.

4.1 A suite to support visual thinking

One of the needs that we extracted in chapter two was that a suite should, in line with the dynamic model cycle, put equal emphasis on analysis, design of alternatives, evaluation, and choice. We reasoned from the position of the user, the airport decision advisor, the domain expert, or the port's area planner. We did not reason from the position of the modeler or the programmer, the person(s) who develop and specify a suite for a specific application domain. In the ABS case we used existing models, glued together by a middleware layer. In the APS case we focused mainly on visualizing information, which may (or may not) be the outcome of an analytical or simulation model. In general, a modeler may choose to use existing models, e.g. the runway capacity model of ABS, build models in existing modeling environments (Matlab, Arena, eM-Plant), or build everything from scratch using a programming language such as C++ or Java.

However, we do not focus on the modeler but aim to support the user of a mainport planning suite. From a conceptual point of view, in terms of the object oriented paradigm, we extend the traditional simulation and modeling suite of conceptualization and specification services (Figure 20). We can call our extension a visualization suite, because we aim to support the activities of analysis, design of alternatives, evaluation and choice in mainport planning visually. We aimed to support visual thinking. In a specific mainport situation, the visualization suite can be extended with domain specific services: services used in a specific airport or seaport such as the RIV system in the PoR, or a service providing access to a customer database. Furthermore services can be added to support the more process management related tasks in mainport planning such as setting agendas and agreeing on actions. It is the whole of visualization services combined with domain specific services which we call a Mainport Planning Suite. In the design of our suite the focus will be on the visualization services, however we should always consider these from the perspective of a Mainport Planning Suite.

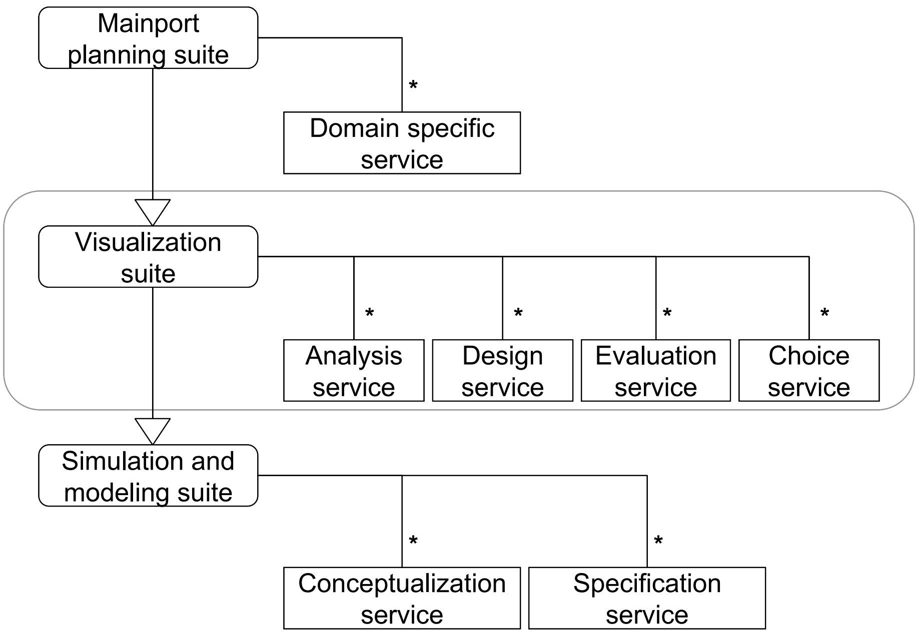


Figure 20: The conceptual relationship between the visualization suite, a simulation and modeling suite and a mainport planning suite.

4.2 Identifying and indexing information

The first step in the visualization process as described by Card et al. (1999) is to gather a set of raw data. Burkhard refers to the knowledge type: know how, why, who, where, what. In mainport planning, the data are distributed over the organization and thus we identified the need for a web-based solution. Berners-Lee (1996) introduced the Universal Resource Identifier, later renamed the Uniform Resource Identifier (URI) to uniquely identify data on the Web: “An object is ‘on the web’ if it has a URI” (Berners-Lee 1996). Today URIs are the standard way of referencing information on the Web, such as webpages, email addresses, file servers, and databases.

The RFC 2396 standards track (Berners-Lee et al. 1998) introduces the concept of an URI in detail. A URI is defined as:

“a compact string of characters for identifying an abstract or physical resource”.

A URI provides a uniform syntax to reference resources. A resource can be anything that has an identity such as electronic documents, images, and services. Although initially invented for identifying resources on the Web, a URI can in principle also be used to identify non-network retrievable resources such as humans, corporations and books. The identifier is an object, consisting of a sequence of characters, which references a resource.

At the highest level a URI is specified by the following syntax[6]:

[*scheme*:]*scheme-specific-part*[#*fragment*]

Where the brackets identify optional parts of a URI. The scheme identifies the specific protocol which applies to the resource. For example http (hyper text transfer protocol) is the commonly used protocol for webpages. The scheme specific part forms the unique address of the resource, and fragment refers to a specific location in the resource. The scheme specific part can be hierarchical:

[//*authority*][*path*][?*query*]

Where the authority part can be further specified as:

[*user-info*@]*host*[:*port*]

Information which can be identified by an URI can be referenced via an index. An example in which information sources throughout a port organization’s departments are referenced is given in Figure 21. For example information regarding locations is managed by the infrastructure department, while shipping routes are maintained by the traffic management department. A Mainport Planning Suite should provide a mechanism for creating an index of the information relevant in a planning project. The process of creating an index is called indexing, which is a process common to search engines. For example, Brin and Page (2000) describe the workings of an indexing system for a large scale search engine, better known as Google®[7]. The indexing system they describe uses a “robot” to follow the links on webpages to generate a queue of URIs to be indexed. Although in mainport planning it is not always clear whether information is available, it is usually possible to identify the required information hosts, such as customer databases and map servers. Therefore, we argued that in

[6] See: http://java.sun.com/j2se/1.4.2/docs/api/java/net/URI.html
[7] Google, Google Maps, and Google Earth are trademarks of Google Inc. in the United States and other countries.

many situations it was possible to resort to simpler indexing systems compared to the one described by Brin and Page. As an example of a simpler approach, consider PhotoMesa (Bederson 2001) which uses a manual approach to indexing, where a user selects folders, on a local hard disk that contain images to be displayed.

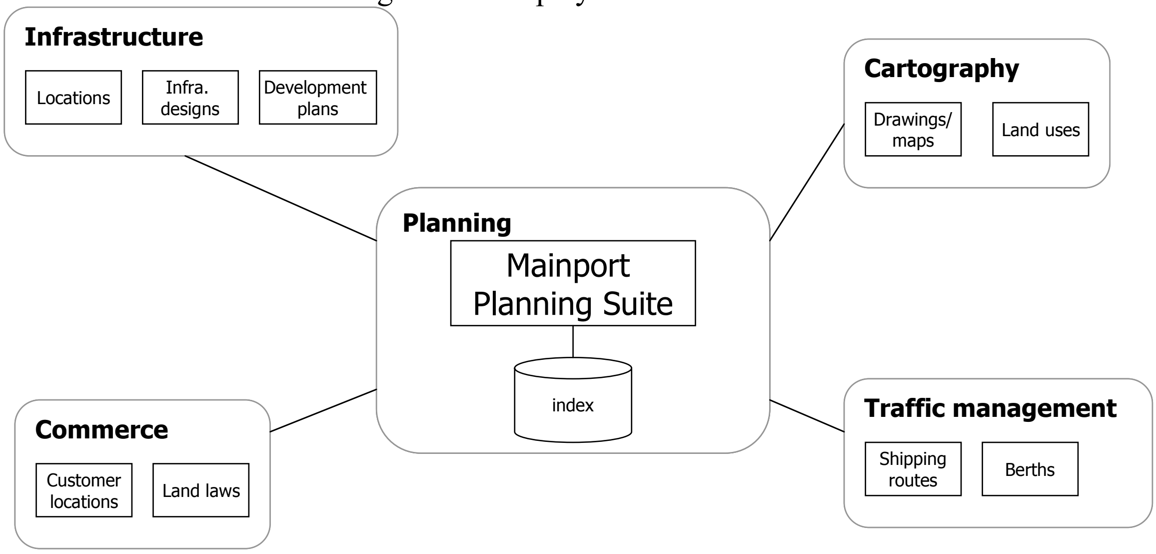


Figure 21: The planning depertment or project team needs to link to and index dispersed information sources within a port organization's departments.

4.3 Structuring, filtering and transforming information

After retrieving and indexing a set of raw data, the data need to be structured, filtered and transformed (Card et al. 1999). This is in line with chapter 2 where we concluded that there is a need to organize the knowledge and information which is used and produced in mainport planning. Information accumulates in the long term leading to large piles of documents, memo’s, maps, designs, pictures, models and so forth. How to organize information is not a new problem, but still people loose track of where they stored their documents, files, and other content. Barreau and Nardi (1995) investigated how people file and find information on computers. They distinguish between three types of searching for files: based on location, e.g. placement on the desktop, logical, in the file system hierarchy, and a text based search, e.g. for filename. They found that users prefer to use location as a reminder of where to find a file. They also found that users keep only limited archived information and concluded that old (historical) information is usually not considered to be useful information. Fertig et al. (1996a) however disagree with Barreau and Nardi’s conclusions and they argue that “if software systems handled archiving and retrieval more conveniently we might find that old information is reused more often”. Furthermore they argue that a location based search may have come out as the preferred method only because no better alternative was available.

More recently Gelernter (2003) has argued that: “operating systems have been traveling away from simplicity”. Information is distributed over the file hierarchies of many computers, PDAs and cell phones: sometimes placed on the desktop, other times in an email box and yet other times in the bookmarks of a browser. He advocates visualizing stored information as a “documentary history”, a sequence of events. Lifestreams (Fertig et al. 1996b) is a solution advocated by Gelernter which implements this concept (Figure 22). It visualizes information as a stream of events ordered on a time axis, showing archived information, and planned documents, events, appointments and so forth.

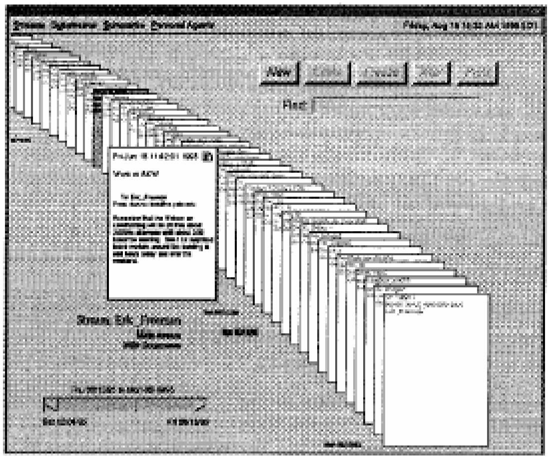


Figure 22: Lifestreams (Fertig et al. 1996b).

Another concept which goes beyond the traditional file system hierarchy is the semantic file system (Gifford et al. 1991). In a semantic file system information is accessed via virtual folders (Jones 1999, Cole et al. 2000). In a semantic file system information is retrieved by queries in indexed file attributes (meta data). Field - value conditions are specified and stored as a virtual folder, which when opened by the user performs a query on the file system using the specified conditions. According to Fertig et al. (1996a) the contribution of the semantic file system is that it can provide many desired viewpoints on a file system.

Furthermore Fertig et al. (1996a) mention Shneiderman's dynamic visual queries (Shneiderman 1994) as a possible new metaphor for visualizing large quantities of stored information. Today however, we see that the software industry is moving in the direction of the semantic file system. Mozilla's®[8] email client Thunderbird™ uses the concept of virtual folders to filter emails on user specified attributes. Apple[9]® already has virtual folder functionality build into their operating system, email and multimedia applications (Figure 23). Microsoft®[10] has followed and introduced virtual folders in Windows Vista®. We argued that we should follow this movement of the computer industry as people will already be familiar with the concept. Hence the use of virtual folders in a mainport planning suite seems to be an appropriate choice for structuring and filtering the information. We aim to archive, or structure searches not the (distributed) content itself. Furthermore, in a distributed, web-based environment we needed to investigate possibilities for shared use of virtual folders. For example users may want to share their virtual folder with the planning team, users with specific rights, or with specified employees, or maybe with everybody who can access the suite.

Our preference for virtual folders did not prevent us from using lifestreams or dynamic visual queries. Information filtered using virtual folders can be visualized as a sequence of events, and dynamic visual queries can be used to narrow down further the search results. For example, we could visually specify a time interval and only visualize the information which

[8] Mozilla and Thunderbird are either registered trademarks or trademarks of the Mozilla Foundation in the United States and/or other countries.

[9] Apple and Mac OSX are either registered trademarks or trademarks of Apple, Inc. in the United States and/or other countries.

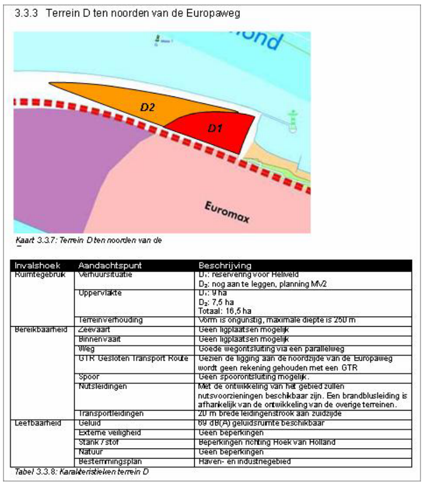
[10] Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

applies to the specified time interval. Furthermore, we could offer a text based search, to search for information in the project database.

Figure 23: Defining virtual folders (Smart Folders) in Mac OSX (source: www.apple.com).

4.4 Semantic information

A semantic file system requires semantic information (meta data) on which to operate. Commonly the semantic information which is stored in a computer's file system is limited to elements such as the file name, date of creation, date of last modification, user and group and permissions. To be able to search based on a richer set of criteria meta data should be added to information. In chapter 2 we identified the need to add annotations to information, keep track of actions and decisions, etc.. An example of annotations made regarding land use in the PoR is provided in Figure 24. In this particular example a map is used to identify the location of two parcels of land and the annotations form a description of the advantages and disadvantages of possible industries at these locations.


3.3.3 Terrein D ten noorden van de Europaweg

Kaart 3.3.7: Terrein D ten noorden van de

Invalshoek	Aandachtspunt	Beschrijving
Ruimtegebruik	Verhuursituatie	D_1: reservering voor Heliveld D_2: nog aan te leggen, planning MV2
	Oppervlakte	D_1: 9 ha D_2: 7,5 ha Totaal: 16,5 ha
	Terreinverhouding	Vorm is ongunstig, maximale diepte is 250 m
Bereikbaarheid	Zeevaart	Geen ligplaatsen mogelijk
	Binnenvaart	Geen ligplaatsen mogelijk
	Weg	Goede wegontsluiting via een parallelweg
	GTR Gesloten Transport Route	Gezien de ligging aan de noordzijde van de Europaweg wordt geen rekening gehouden met een GTR
	Spoor	Geen spoorontsluiting mogelijk.
	Nutsleidingen	Met de ontwikkeling van het gebied zullen nutsvoorzieningen beschikbaar zijn. Een brandblusleiding is afhankelijk van de ontwikkeling van de overige terreinen.
	Transportleidingen	20 m brede leidingenstrook aan zuidzijde
Leefbaarheid	Geluid	69 dB(A) geluidsruimte beschikbaar
	Externe veiligheid	Geen beperkingen
	Stank / stof	Beperkingen richting Hoek van Holland
	Natuur	Geen beperkingen
	Bestemmingsplan	Haven- en industriegebied

Tabel 3.3.8: Karakteristieken terrein D

Figure 24: A map with annotations about possible used of parcel of land (Gebiedsplan Noordwesthoek, PoR 2005).

Today making annotations on the Web is often referred to as blogging. A blog is an online diary, a frequently updated webpage with posts in a typically reverse chronological order (Schiano et al. 2004). Blogging can be considered as a new form of expression and communication which has gained an unprecedented popularity during the past few years. Schiano et al. mention that bloggers, as they are called, write about daily life, travels, family, events, politics and many other subjects.

As described by Schiano et al. the first blogs appeared in 1997. At that time no software tools were available to aid in setting up and maintaining a blog. Blood (2004) explains that weblogs preceded web blogs. Weblogs were a log of links to other sites, usually accompanied by a short description and personal thoughts. In 1999 Blogger® (www.blogger.com) was one of the first companies to launch web blogging software, which enables people to publish content at the push of a button.

Blogging software, wiki software (Leuf and Cunningham 2001) and online bulletin boards (e.g. phpBB) have developed fast over the last decade, leading to innovations such as: the permalink, comments and trackbacks (Blood 2004). Modern blogging software also offers additional functionalities such as user registration, calendar, e-mail to members, polling, topic categorization, and uploading or downloading images (Mongkolwat et al. 2005). Furthermore, specialized types of blogs have appeared such as photo blogs where users can add annotations to photos and images (Cohen 2005). In 2004 a new phenomenon arose called Podcasting, which is a form of blogging that uses sound recordings instead of text.

Cayzer (2004) talks about semantic blogging. He sees blogging as an ideal way to support decentralized, informal knowledge management: "… blogs can publish machine-readable summaries of their content, allowing individual or community aggregators to collect, merge, sort and index this data." These machine readable summaries (feeds) contain meta data which is often specified in either the Really Simple Syndication (RSS), or Atom format (Nottingham and Sayre 2005). These XML based formats can automatically be parsed by for example news readers and modern web browsers. Below is an example of an Atom feed:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">

  <title>Example Feed</title>
  <link href="http://example.org/"/>
  <updated>2003-12-13T18:30:02Z</updated>
  <author>
    <name>John Doe</name>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>

  <entry>
    <title>Atom-Powered Robots Run Amok</title>
    <link href="http://example.org/2003/12/13/atom03"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2003-12-13T18:30:02Z</updated>
    <summary>Some text.</summary>
  </entry>

</feed>
```
(source: http://www.atomenabled.org/developers/syndication/#sampleFeed)

As is shown in the example an Atom feed contains meta data such as the feed's title, a URI identifying the root location of the feed, information on when the feed was last updated, information about the author and a URI specifying a unique name of the feed. Next the feed can contain a number of entries, which specify the actual information resource that is annotated. Each entry contains similar items as before such as a title, URI, date, summary. The provided example shows only a subset of the available Atom tags. In addition Atom's vocabulary can be extended with elements from other namespaces. One of the blogging systems which uses the Atom format is the previously mentioned Blogger.com.

As mainport planning is a long term process where the composition of the planning team changes over time, the use of RSS or Atom feeds can help the involved actors to scan quickly for information updates that are relevant for them. For example, a commercial expert in the PoR is interested in which land parcels will become available to new customers. He or she therefore may want to be alerted when a planning team decides to change the layout of land parcels in a specific port area. He or she may not be present at the planning meeting in which such a decision is made, and then a news feed is a useful service in such a situation.

4.5 Creating visualizations

To create visualizations we need to map visual structures onto the transformed data (Card et al. 1999). As we learned from chapter 2, the information in mainport planning can be of virtually any type and the information needs of the involved actors differ substantially. Hence, we cannot predefine a specific visualization type. Some information may be adequately represented as a diagram, while other information may be better shown in a realistic animation.

A common denominator is the shared viewpoint: the geographical map of the area under investigation. Planners use the map to gain insight and understanding in the available infrastructure, facilities and resources; to find possibilities, constraints and bottlenecks; and to make sketches of alternative solutions. Or as Steenbergen et al. (1999) formulate it: “Every drawing is in fact an answer to a question; in answering one, it raises the next. … Each drawing evokes a subsequent one in a continuous process of new questions, findings and attempts (page 158).” Alternative solutions and sub solutions can be layered or put in a table of small multiples (Tufte 1990), or a morphological chart (Cross 1994) such that they can be compared. After comparing the alternatives they should be evaluated, which can be visualized using e.g. objectives trees, or evaluation charts (Cross 1994) (Malczewski 1999).

We could make use of the following options to create visualizations.

- Using low level libraries such as Java2D and Java3D to create visualizations from scratch
- Using high level libraries such as Geotools to create visualizations by using standard components
- Using an existing application or service to visualize the information

The last option has the advantage that no additional software needs to be developed. Furthermore users may already be familiar with the application or service. They might even have created the information in the application. For example, a spreadsheet application enables users to enter data and to apply visual structures to create visualizations. In such case it may not be desirable to replicate this functionality in an external service.

In other situations the original application in which the data were produced may be to complicated, or just not fit the intended audience of our Mainport Planning Suite. For example, the RIV system in the PoR turned out to be too complicated for use in area planning projects. However, RIV contains geographical data that are specified according to industry standards e.g. following the standards of the OpenGIS Consortium, thus it became possible to use an off the shelf component library to render the data contained in RIV. For example Geotools (www.geotools.org) is an open-source implementation of the OpenGIS standard. This viewer can be fully tailored to the needs of a mainport planning team. Only when no suitable component library is available should one resort to developing a visualization from scratch.

4.6 Interactive views

Visualization tools and techniques will eventually end up in graphical user interfaces (GUIs). Modern graphical user interfaces have their roots in the research of the 1970, 1980s and 1990s (Carrol 2001). Today's window managers and the graphical user interfaces found in e.g., MS Windows®, Mac OSX® and Linux, have their origins in the research at Stanford Research Institute, Xerox Palo Alto Research Center (PARC), and MIT in the 1970s. Myers et al. argue that the inventions done during those years shaped the current "GUI style", and the development of event languages, interactive graphical tools, component systems, scripting languages, object oriented programming and even the use of hypertext.

With the rise of the Internet graphical user interfaces are moving onto the Web. Webpages are no longer static documents, but have become full fetched graphical user interfaces. Graphical user interfaces are distributed clients for programs that run over a network of computers within an organization and between different organizations. In this section we describe the alternative software architectures for supporting the creation of interactive views on distributed information.

4.6.1 The MVC framework

Modern GUIs are constructed of user interface components such as windows, buttons, text boxes and slider bars. Key to the workings of a GUI is the Model-View-Controller (MVC) framework, which was first introduced in the Smalltalk programming environment (Goldberg 1983). Later on Shan (1989) improved this framework by introducing an event-driven MVC framework; the original concept was based on a polling mechanism.

The MVC framework consists of three main components: models, views and controllers. Each of these components represent software objects which fulfill a specific task in the GUI. As Krasner and Pope (1988) describe: "Models are those components of the system application that actually do the work. … They are kept distinct from views which display aspects of models. Controllers are used to send messages to the model, and provide the interface between user interface devices (e.g. keyboard and mouse)."

A GUI consists of one or more views, where a view represents a rectangular area on the screen. An important aspect of GUIs is that views are designed to be nested (Burbeck 1992). For example, a screen can display multiple windows, which each consist of user interface components (i.e. views) such as buttons and text boxes. Windows, buttons and text boxes are all examples of views and e.g. buttons are nested inside windows, i.e. there exists a tree, a hierarchical relationship, between the various components of a GUI.

This hierarchical relationship is key to understanding the inner workings of the MVC framework. User interaction, by mouse or keyboard, always targets a specific coordinate on the screen. This coordinate can be part of one or more views. Yet, the user meant to target a specific view, e.g. he or she wanted to click a button. As Burbeck explains, the trick is to find a controller which handles the user input for that coordinate. This is achieved by transferring through the tree until the top most GUI component is found at the specified coordinate. Next the user event, e.g. clicking the mouse, is passed to the respective controller, which calls on the model to execute the application logic and uses the outcomes to update the view.

Today many different toolkits and libraries exist, e.g. Java Swing® and Visual Basic®, which support the development of desktop GUIs following the MVC framework. Hence, developing desktop applications according to the MVC framework has become common practice.

4.6.2 User interfaces in classic web applications

While the MVC framework was commonly applied in the design of desktop applications, this concept again gained attention in the development of web applications. When the Internet had just been introduced to the general public in the mid 90s its content consisted mainly of static HTML pages. The only type of 'button' found on a static HTML page was a hyperlink which brought you to a next page (Berners-Lee and Connolly 1995). Soon people realized that the Internet had a much larger potential if entire interactive applications could be moved onto the Web (Nielsen 1999). This however, raised new challenges with respect to user interface design because this meant that webpages became GUIs for web applications.

A main challenge was that webpages were designed to represent documents, not full fletged user interfaces. Somehow the gap between the limitations of HTML and modern GUIs needed to be bridged. A further challenge is to be found in the distributed nature of the Internet. As described by Alonso et al. (2004), web applications are often based on an N-tier architecture in which three (horizontal) main layers of functionality can be distinguished: the presentation layer, the business logic layer, and the resource management layer. The resource management layer manages the underlying (persistent) data; the business logic layer does the processing; and the presentation layer renders the user interface. Each of these layers can be physically distributed to run on different servers. Apart from this horizontal separation, which leads to a three-tier architecture, web applications can also use a vertical separation. This means that web applications are distributed into functional units which offer services to each other over a network (Endrei et al. 2004). Alonso (2004) explains that this horizontal and vertical decoupling comes with a performance penalty because the different servers have to communicate over a network, but the advantage is increased maintainability and scalability.

The question is how to fit the MVC framework onto an N-tier architecture for a web application? Crane et al. (2006) make an extensive analysis of the presentation layer of web applications. A classic web application is a pure server side application, which means that the client is often a web browser (Figure 25). The web browser does nothing else other than displaying the user interface (a webpage) and sending user inputs back to the web server. Hence the workflow of a classic web application is (Garrett 2005):

- send a GUI to the web browser (HTML and CSS)
- receive user inputs (http requests)
- process the user inputs
- send an updated GUI to the web browser and wait for new user inputs etc.

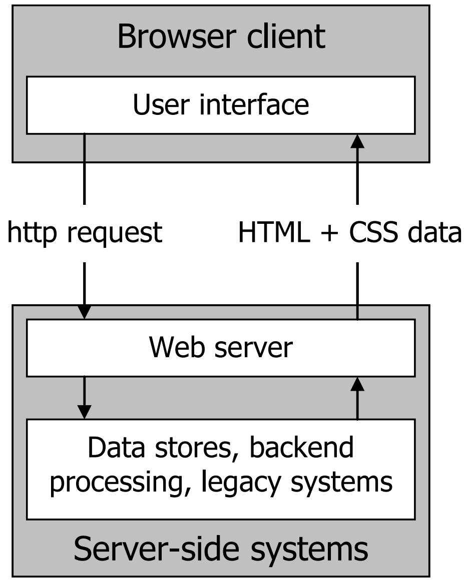


Figure 25: A classic web application (Garrett 2005).

The GUI of a web application remains a webpage, however: HTML elements are not user interface components like those in a desktop application. In order to facilitate designers of web applications, several vendors and open source communities have developed software libraries with standard MVC components. These components mimic the structure found in desktop applications: GUI components have a server side representation in code while they are painted as HTML elements in the browser, and user interactions are handled by event listeners (i.e. controllers). On a low level techniques such as Java Server Pages (JSP) were introduced to decouple application logic and graphical layout, or in other words: to bridge the gap between programming code and HTML code. On a higher level libraries such as Apache[11] Struts and Java Server Faces (JSF) were introduced to implement a full server-side MVC workflow (Farley and Crawford 2005). This is illustrated in Figure 26 where a client interacts with a web server:

- a controller generates views (i.e. webpages) which are sent to a client
- client inputs are received by the controller
- the controller delegates the inputs to action handlers
- the action handlers call upon the business logic (i.e. model)
- the controller updates the view etc.

It should be noted that the business logic (models) may reside on the same webserver as the views and controllers, or it may be distributed among a cluster of application servers. The former represents a classic web application, while the latter represents a full N-tier architecture as described by Alonso et al. (2004). However, this makes no substantial difference for implementation of the MVC framework. The MVC framework implementation may run on a cluster of distributed servers (e.g. Apache Tomcat servers), such that load balancing can be applied to improve scalability (Farley et al. 2006, Genender et al. 2006).

[11] Apache is a trademark of The Apache Software Foundation.

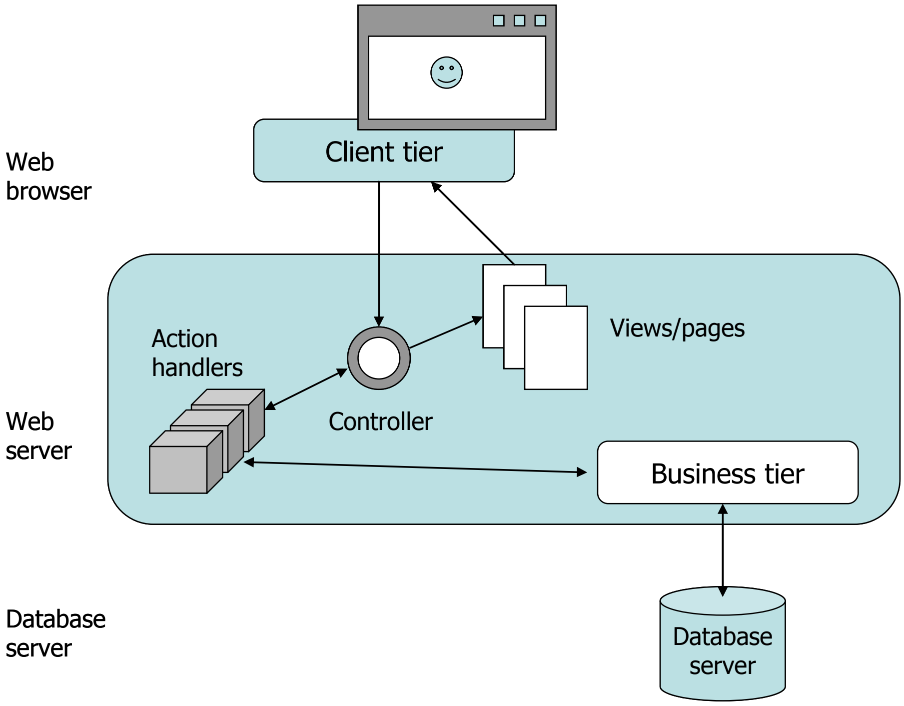


Figure 26: The MVC framework in a classic web application (Crane et al. 2006).

4.6.3 Rich-web based clients

The MVC workflow increases the scalability and maintainability of the presentation layer, however another problem with browser based clients is that all interaction happens via the web server. The browser acts as a 'dumb client' which simply downloads and displays the user interface and sends all interaction requests back to the server to be processed. Nielsen (1999) made a careful prediction that the Web would mainly be used for browsing information and lightweight interactions. However, as described by Crane et al, lightweight interactions such as authorizing users are no longer sufficient. The demand for a richer user interaction has started to come in conflict with technical capabilities. Browser based clients could not replicate the rich user experience of desktop applications: "Web users are getting tired of the traditional web experience. They get frustrated losing their scroll position; they get annoyed waiting for refresh; they struggle to reorient themselves on every new page." (Crane et al. 2006) Instead of sticking to lightweight interactions, the Web has increasingly become a fully interactive experience.

To overcome the limitations of browser based clients several solutions are available. Sun Microsystem's Java Web Start®[12] is a specification for bundling web applications on a webserver, such that a desktop process can find, download and execute them (Crane et al. 2006). Applets are Java programs which can run inside a web browser and can thereby provide a rich user interface where user interactions are processed (at least partly) on the client. Similar approaches are followed by Microsoft's ActiveX®[13] and by Adobe's Flash®[14] which also provides extensive animation capabilities. A disadvantage is that the client must have the libraries and plugins installed to execute e.g. Applets and Flash programs.

[12] Sun, Sun Microsystems, the Sun Logo, Java, and Java Web Start are either registered trademarks or trademarks of Sun Microsystems, Inc. in the United States and other countries.
[13] Microsoft and ActiveX are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
[14] Flash is a trademark or registered trademark of Adobe Systems Inc. in the United States and other countries.

In 2005 Garrett (2005) coined the term Asynchronous Javascript And XML, acronym AJAX: a development technique, or recipe for creating rich browser-based clients using existing web technologies. The goal of AJAX is to make webpages more responsive by exchanging small amounts of data between the client and the server, instead of reloading an entire page whenever user interaction takes place. Following AJAX, the browser hosts an application not content, and the server delivers data not content (Crane et al). In other words, a web server sends a browser-based web application, a dynamic web page, to the browser. Next, this web application responds to user interaction by producing asynchronous data requests to the server (Figure 27).

As an example of the potential of rich clients in mainport planning consider Google Maps™, which offers a web-based and interactive geovisualization in the public domain. Google Maps follows the AJAX paradigm and therefore requires no browser plug-ins such as Flash or Java libraries. Yet it provides a fully interactive and searchable map of the world with a striking level of detail when fully zoomed in. Depending on the zoom level and screen panning a map, satellite image, or combination of both is rendered on the server and sent to the client. Markers are rendered on the map to indicate the location of search results. Furthermore it is possible to integrate Google Maps in (publicly available) third party webpages and to show custom layers and markers on the map.

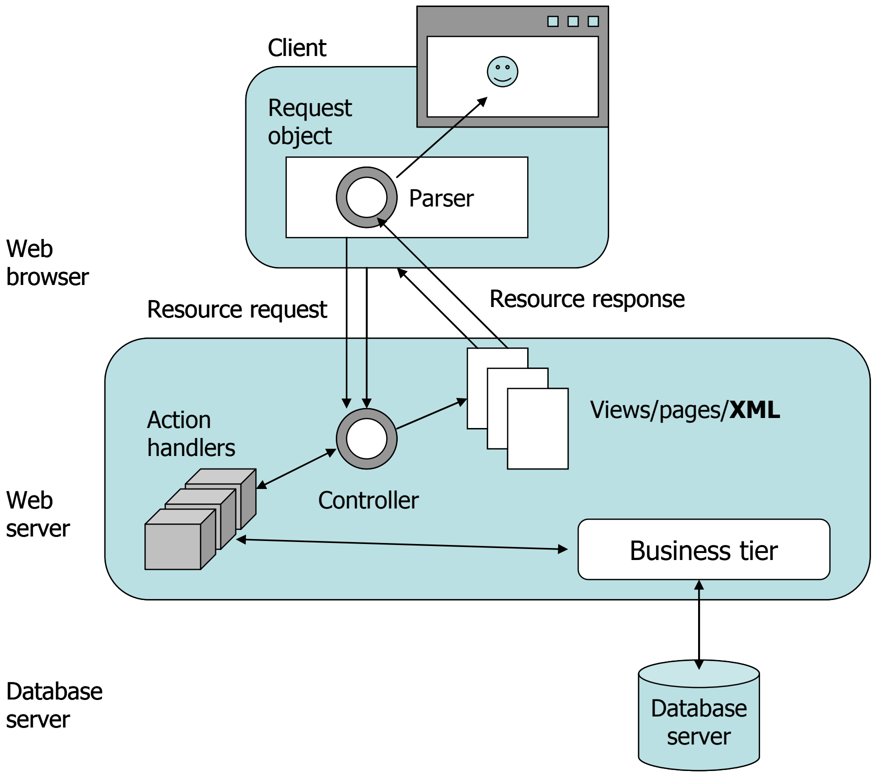


Figure 27: An Ajax client.

4.7 Web portal: aggregating views for users in different roles

An organizational chart of a typical airport organization is displayed in Figure 28. A planning department develops strategic plans and provides these to e.g. the commercial department, engineering department and the operations department.

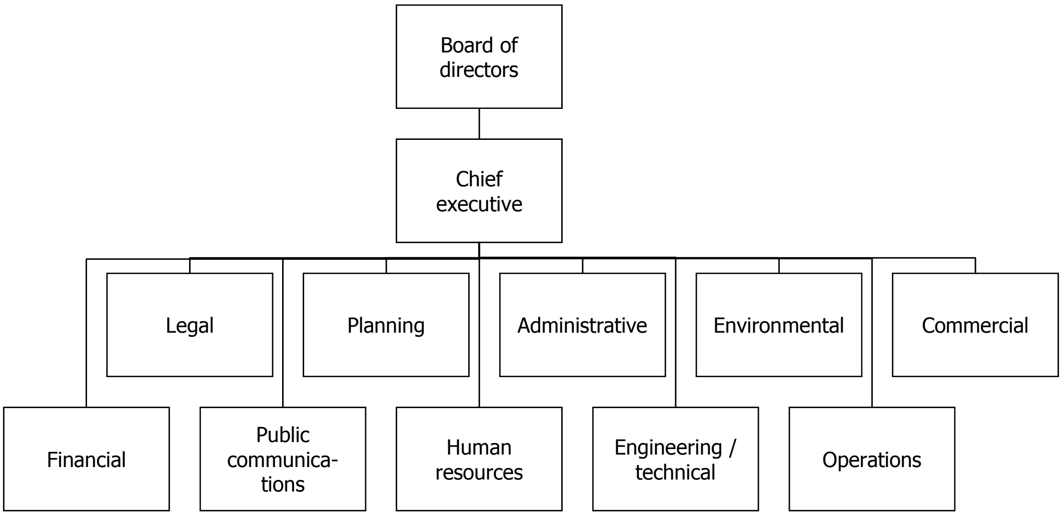


Figure 28: An organizational chart of an airport (Neufville and Odoni, 2003).

Mainport planning involves actors in many different roles, who are therefore interested in different types of information, or who may prefer to view the same information from a wide variety of perspectives. A web application provides us with the ability to distribute our graphical user interface over a network. An enterprise information portal, or web portal is a web application which "commonly provides personalization, single sign-on, content aggregation from different sources and hosts the presentation layer of Information Systems" (Abdelnur and Hepper 2003). In an enterprise information portal disparate information is consolidated in views, that are configured, or personalized for actors in different roles. For example Liferay Portal®[15] provides a hierarchy of administration scopes, making it possible to create and manage portal pages that fit the needs of organizational units as well as those of individual users (Figure 29).

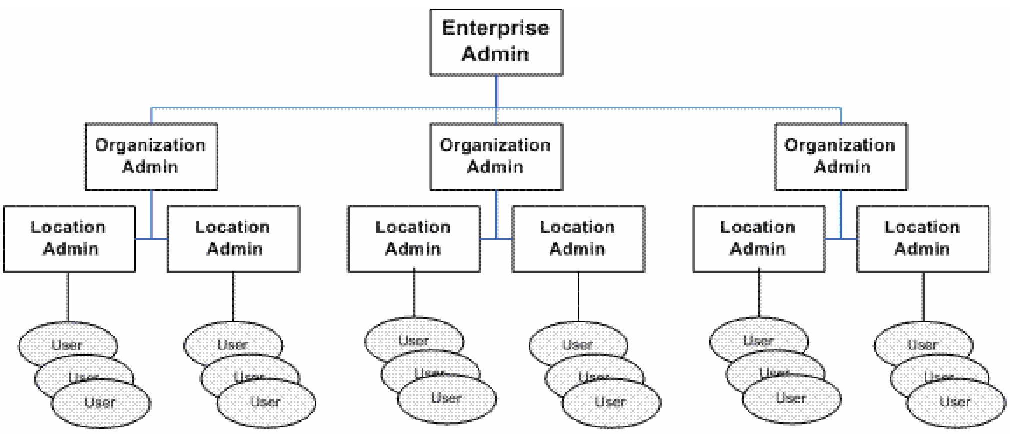


Figure 29: Scopes of administration in Liferay Portal (www.liferay.com).

[15] Liferay Portal is a trademark of Liferay, Inc.

Furthermore, content aggregation means that the page produced by a web portal consists of pluggable user interface components called portlets. A portlet generates a fragment: a piece of markup (e.g. HTML) that is aggregated with other fragments to generate a complete page. Portlets provide interactive views on information following an MVC workflow framework, which are offered to users with specified roles (Figure 30).

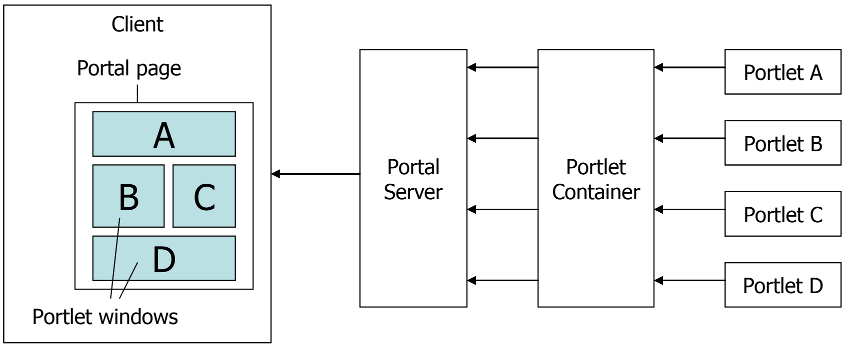


Figure 30: Portal page creation (Abdelnur and Hepper 2003).

Abdelnur and Hepper (2003) describe JSR168, an interface standard for portlet applications, which is widely supported by the software industry and open-source projects. This standard eliminates a lock in on a specific portal. For example, a portlet application that was initially developed and tested on an Apache Jetspeed portal, can now be deployed on a Liferay portal without having to change a single line of code. In that sense we can regard portlet applications as web services (Alonso et al. 2004, Papazoglou and Dubray 2004) in a Service Oriented Architecture (SOA) (Endrei 2004). A portlet application offers a specific service, which is hosted by a service provider, which can be discovered through a service broker, and which can be used by a service client. Suites of services can be created, configured for the situation at hand and users in specified roles, and made available through a portal.

We conclude that an enterprise information portal provides us with an appropriate a platform for hosting a mainport planning suite. The services provided by the Mainport Planning Suite can seamlessly be aggregated with existing services into personalized views for users in different roles.

4.8 Requirements for a Mainport Planning Suite

In chapter one we stated our research question:

Can we create a suite of services to support the actors in a studio based planning process that improves their effectiveness in mainport planning?

We argued that such a suite would support invoking the memory and the creativity of multiple actors, with different objectives, who were specialized in different fields of knowledge, and who worked in different contexts, such that effective mainport planning can be conducted.

On a general level we adhere to Sol (1982) in that such a suite requires metatheoretical freedom, conceptualization freedom, modeling freedom and solution finding freedom. However, in this chapter we have provided argumentation for our decision to focus specifically on the activities of analysis, designing alternative solutions, evaluation and choice. These activities mainly refer to solution finding freedom, whereas conceptualization and modeling freedom refer mainly to the activities of conceptualization and specification which are extensively addressed by e.g. Jacobs (2005). Furthermore we have shown that distribution is an important requirement of such a suite.

Keen and Sol (2007) state that effective decision support rests on three Us: usefulness, usability and usage. Usefulness expresses the added value of tools and methods to the decision making process. Usability expresses the mesh between people, process and technology, and finally usage expresses flexibility, adaptivity and suitability to the decision making context. In other words, usefulness expresses the added value of our suite, usability expresses ease of use of our suite, and usage expresses how the suite can be integrated in a mainport's organizational and IT infrastructure. The thee Us provided us with a theoretical basis for working out general requirements into a set of detailed requirements for such a suite. In the following subsections we introduce the requirements for a Mainport Planning Suite following the 3Us.

4.8.1 Usefulness

Our suite should support inter-disciplinary teams of mainport planners to analyze mainport areas, design alternative solutions for mainport areas, evaluate these alternative solutions for relevant aspects, and to make a choice for a preferred solution. As we observed during our case studies, these activities require mainport planners to make trade-offs regarding aspects such as available space, accessibility, livability, sustainability and commercial attractiveness. To make these trade-offs they require insight and understanding into how different objects or concepts are located relative to each other. Furthermore mainport planners have to deal with dynamic information e.g. how the area changes over time. This brings us to the following requirements:

Requirement 1. A Mainport Planning Suite should provide visualization services to support analysis, design of alternative solutions, evaluation and choice in mainport planning projects.

Requirement 2. A Mainport Planning Suite should Support the visualization of temporal and location based information.

In chapter 2 we concluded that a geographic map is used by mainport planners as a shared viewpoint and a writing pad. During a mainport planning process maps, which are prepared by GIS analysts, are used to analyze, evaluate and communicate about design alternatives. A geographic map can be considered as a specific representation of location based information, which is based on GIS technologies e.g. geographic projections and shape files (Slocum et al. 2005). Therefore support for geovisualizations is a requirement of our suite.

Requirement 3. A Mainport Planning Suite should support the visualization of geographic maps.

In mainport planning an abundance of information of many types is used and produced. In chapter 3 we adhered to Card et al. (1999) in that we want to make use of "computer-supported, interactive, visual representations of data to amplify cognition." Based on the principles of visualization, we identified promising technologies for indexing, filtering, viewing and interacting with information. Consequently we specified the following requirement:

Requirement 4. A Mainport Planning Suite should support indexing, filtering, viewing and interacting with the different types of information used and produced in mainport planning.

During our case study in the PoR we observed that mainport planning projects take months of work. During this period the mainport planners have meetings in which they discuss and comment on each others ideas. Furthermore they need to agree on and keep track of actions and decisions. In chapter one we adhered to Simon (1986) who stated that planning is strongly intertwined with problem solving and decision making. Where decision making deals with activities such as analysis, design evaluation and choice (requirement 1), problem solving deals mainly with fixing agendas, setting goals and designing actions. Hence we specified the following requirement.

Requirement 5. A Mainport Planning Suite should support the making of annotations, comments, and keeping track of actions, decisions and meeting minutes.

4.8.2 Usability

Visualizations are provided through the GUIs of our suite. The design of a GUI determines for a large part the usability of our suite. Shneiderman (1998) summarizes the main rules which should be followed in the user interface design of software applications. We adhere to the eight golden rules of Shneiderman to address the usability of the GUIs of our suite:

Requirement 6. A Mainport Planning Suite should follow the eight golden rules of user interface design (Shneiderman 1998).

In this chapter we argued that we require a web-enabled suite which is provided through an enterprise information portal, i.e. a web portal. In addition we argued that technologies such as AJAX should be used to overcome the usability limitations of the “thin clients” offered by classic web applications. Considering that the web portals of today are still built as classic web applications, providing rich clients through a web portal is a challenge which we should address in the design of our suite:

Requirement 7. A Mainport Planning Suite should provide rich clients for user interaction.

Requirement 8. A Mainport Planning Suite should be provided through a web portal.

We argued that our suite should be designed based on loosely coupled services for visualization and interaction (research question 3). A web portal offers functionality to make services and information available based on a user’s role; as such it provides an architecture which is suitable for designing loosely coupled services for visualization and interaction. However, we should note that a web portal’s architecture facilitates decoupling of services, but it does not enforce it. Therefore we state the following requirement:

Requirement 9. A Mainport Planning Suite should provide loosely coupled services for visualization and interaction.

4.8.3 Usage

Requirements regarding the usage express how our suite should be integrated with the IT infrastructure and the organizational infrastructure of a mainport. Considering the flexibility, adaptivity and suitability of our suite we argue that the suite should be designed to make optimal use of existing, proven and possibly standardized IT building blocks. This requires a careful partitioning of our suite into loosely coupled subsystems, which leads us to the following requirement:

Requirement 10. A Mainport Planning Suite should be partitioned in loosely coupled subsystems which are based on proven and standardized IT building blocks, and which are specified using clearly documented interfaces.

To address the integration of the suite in the organizational infrastructure of a mainport we specified the following requirements:

Requirement 11. A Mainport Planning Suite should be introduced in the mainport organization with policies on ensuring the completeness, quality and punctuality of information.

In the following two chapters we address these requirements in a design for a Mainport Planning Suite.

5 Building blocks for a Mainport Planning Suite

In chapter 4 we formulated the requirements for a suite to support mainport planning. Using these requirements, the literature reviews in chapters 1 and 3, and the outcomes of the exploratory case studies in chapter 2, we present the design of a Mainport Planning Suite (MPS). Hereby we address our third research question:

Can we design and implement a suite to support studio-based mainport planning using an architecture which is based on loosely coupled services for visualization and interaction?

We first introduce an overview of our MPS. Next we split the description of our design into two parts which we call the technical design and the functional design. In this chapter we present the technical design and in the following chapter we present the functional design. The technical design focuses on the foundations for a MPS: the building blocks, or software components and code libraries, needed to create the services in the suite. The functional design focuses on the services offered by the suite after assembling the building blocks.

In 5.1 we introduce the design of our MPS. Next we choose a programming platform in 5.2 which forms the foundation for the suite. In 5.3 we identify the required building blocks which meet the specific requirements for our suite. In 5.4 – 5.6 we present the detailed design of the most important building blocks of the suite.

5.1 A suite to support mainport planning

In Figure 31 the relation between suites, studios and the business architecture is shown as introduced by Keen and Sol (2007). As we explained in chapter 1, suites are used to provide the technology to support studio based mainport planning. Following Keen and Sol, studios fit into an architecture of "companies with business goals, organizational infrastructures, financial systems and many others."

We divided the three circles with a horizontal line. This line marks the boundary between the business side and the technology side. This distinction was also reflected in our requirements with respect to usage, which address the IT infrastructure and the organizational infrastructure (requirements 10 and 11).

The business side represents the organizational and social aspects of studio based decision making. It shows the environment in which a MPS is used. For example, during our case in the PoR (chapter 2) we saw that mainport planning is conducted in the form of planning projects, which consist of a network of actors, discussion agendas, planned actions, project planning etc.. A MPS is intended to be used inside these mainport planning projects. Mainport planning projects, in their turn, are conducted inside the organizational infrastructure of the mainport.

The technical side of the figure represents the building blocks and IT infrastructure which are used to support the business side. A MPS is deployed in a corporate IT infrastructure. For example, the IT infrastructure of the PoR consists of an intranet, ERP systems, GIS systems, financial databases, e-mail servers, e.t.c.. Relevant information in these systems should somehow be channeled to the different actors involved in mainport planning projects. In chapter 4 we argued that enterprise information portals can be used as an enabling technology to aggregate content and visualize information from different sources for users in different roles.

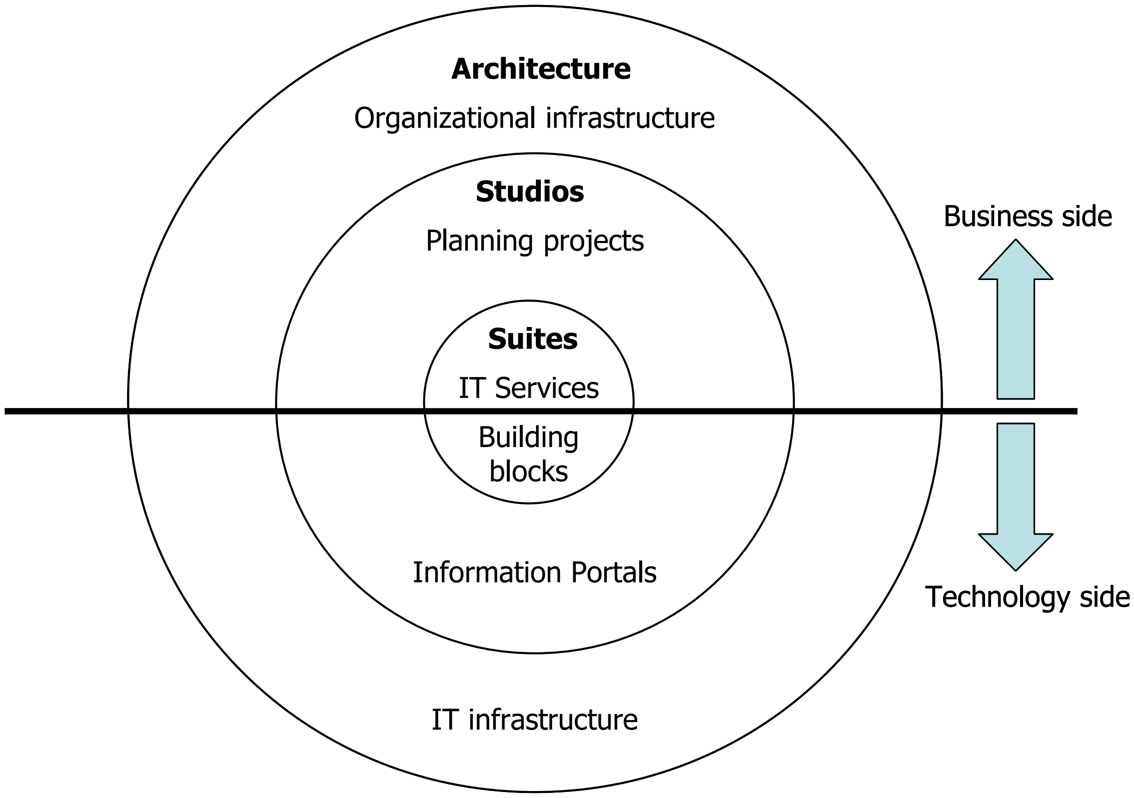


Figure 31: A mainport planning suite and its relation to studio based planning (based on Keen and Sol 2007).

In mainport planning projects, the suite forms the link between the business side and the technical side. It offers IT services to the people in mainport planning projects. These services are assembled from building blocks: software components which provide the required functionalities.

Figure 32 shows another view on the role of a MPS. The mainport planning studio is not at one location, but is dispersed over meeting rooms and offices in the mainport organization. In that sense a MPS differs from e.g. a television or recording studio: the mainport's meeting rooms are not standard equipped with tools that support mainport planning. Instead, when a mainport planning team occupies a meeting room they transform the room into a studio. A MPS changes a meeting room into a mainport planning studio in a digital way: it offers services, tools, documents, maps, electronic agendas and so forth, through an enterprise information portal which hooks up to the corporate IT infrastructure. A MPS sets up a dynamic network of IT services through which the involved actors can access, edit and share information. It offers services for analyzing mainport regions, designing creative solutions, evaluating and choosing preferred solutions.

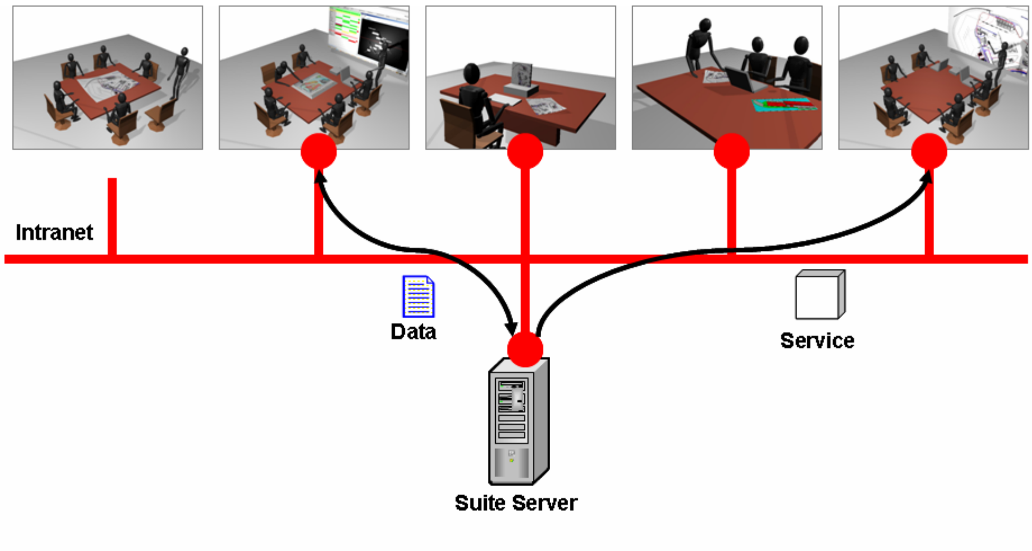


Figure 32: The mainport planning studio consists of a network of temporary nodes.

5.2 Choice for a programming platform

We continue this chapter with a detailed description of our MPS design. We focus on the designing of the technical building blocks from which we can assemble the suite's services, followed by a description of the design of the services in chapter 6. We first had to choose a programming platform[16], which fits the IT infrastructure of a mainport and meets the requirements for our suite. A programming platform describes the software framework that facilitates software development. It usually consists of a programming language, a software architecture, code libraries, and support tools. Ideally we would have liked to choose a programming platform that provided us with a set of building blocks for our design, which could directly be assembled into useful mainport planning services (requirement 10). In reality, however, we had to deal with the current state-of-the-art.

We chose Java Platform, Enterprise Edition (Java EE) as the main programming platform for our MPS. The Java programming language offers the advantage of a strict object oriented specification and thereby supports the de-facto programming style of today. Java comes in two "flavors": the standard edition and the enterprise edition. In contrast to the standard edition the enterprise edition provides middleware[17] for building web applications using an N-tier architecture. Sun Microsystems explains that:
"Building on the solid foundation of Java SE, Java EE provides web services, component model, management, and communications APIs that make it the industry standard for implementing enterprise class service-oriented architecture (SOA) and Web 2.0 applications."
Therefore we concluded that Java EE suited our requirements as presented in chapter 4.

[16] The term platform is commonly used in IT however it is usually not explicitly defined. Often it refers to operating systems e.g. when stating that software is "cross-platform". We, however, refer to a programming platform and thereby follow the terminology as used by Sun Microsystems (see http://java.sun.com/javaee/). A programming platform is usually more than just a programming language. It includes code libraries, a software architecture, and support tools. Where Sun Microsystems uses the term programming platform, Microsoft Corp. uses the term programming framework, e.g. in ".Net Framework".
[17] As defined by the ObjectWeb Consortium: "In a distributed computing system, middleware is defined as the software that lies between the operating system and the applications on each site of the system." (http://middleware.objectweb.org/)

Farley and Crawford (2005) explain that Java EE provides the building blocks for distributed computing technologies and services. They state that it is important to understand that J2EE not only provides the tools and APIs for developing enterprise applications, it also provides a model for assembling and deploying the components of applications. Components are functional elements of an application that run in a container. A container is a service that provides runtime services related to e.g. security, lifecycle management, transaction management (Farley and Crawford 2005).

An example of a container relevant for the design of our MPS is a servlet container. A servlet is an object used for the generation of dynamic HTML content (Farley and Crawford 2005). Adhering to the MVC framework as introduced in chapter 4, a servlet can be used as a controller (Seshadri 1999)[18]: it receives user inputs from the client (called requests), delegates these requests to the business logic, and updates the GUI of the client accordingly (called response). Furthermore servlets are persistent between invocations, which means that they keep their state between user interactions. Figure 33 shows an example of using servlet containers in a scalable distributed architecture (Genender et al. 2007). The rectangular boxes represent servers, and the lines represent network connections. A web server is used to provide content to the client. If a client requests access to dynamic content, i.e. a web application, the web server delegates the request to a servlet container (server) which in its turn delegates the request to the responsible servlet. Next the servlet can call upon business logic provided by application servers or handle the request locally.

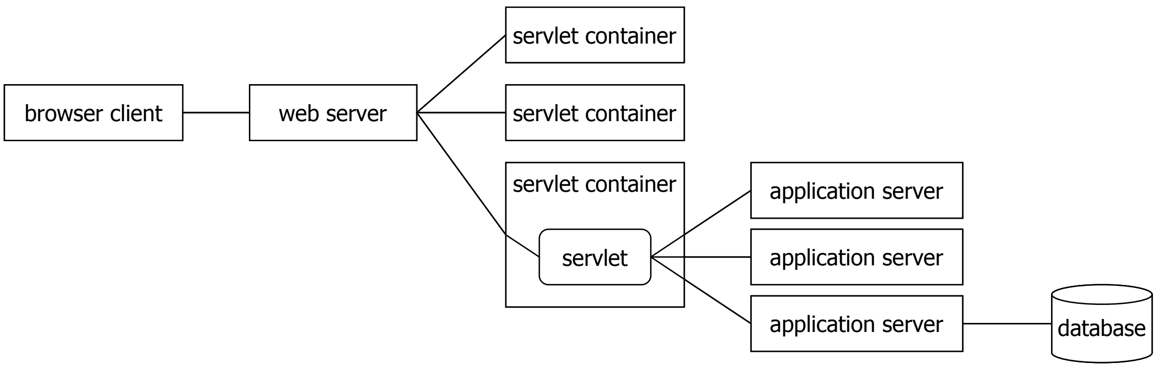


Figure 33: A servlet container in an N-tier architecture.

Java EE is an appropriate programming platform for our MPS design, but in chapter 4 we also introduced the requirement that our MPS should be provided through a web portal, i.e. an enterprise information portal. Following the definition of Abdelnur and Hepper (2003): "a portal is a web based application that –commonly- provides personalization, authentication, content aggregation from different sources and hosts the presentation layer of information systems." In terms of technology a web portal implemented on Java EE is a web application that makes use of servlets (Abdelnur and Hepper 2003), i.e. a web portal runs inside a servlet container.

[18] Note that we regularly reference web articles and technology websites in this chapter instead of publications which are published through the regular academic channels (e.g. journals). We found that state-of-the-art IT technologies are commonly first introduced through recognized technology websites, blogs and wikis.

Abdelnur and Hepper (2003) introduced the Java™ Portlet Specification, or JSR168, which is a standard for developing portlet applications. Portlet applications consist of one or more portlets. A portlet provides a specific piece of content, called a fragment, which is to be included as part of a portal page (Abdelnur and Hepper 2003). Commonly a fragment is a piece of dynamic HTML code which represents a (part of a) web-based GUI. The web portal is responsible for aggregating fragments into a complete portal page. Portlet applications that are designed in compliance with JSR168 are guaranteed to run on any web portal which makes use of the Portlet API. The Portlet API is a library of Java objects and interfaces which implement the Portlet Specification. To meet our requirement to provide our suite through a web portal, we chose to adopt the Portlet API as part of our programming platform (Figure 34). In other words, our MPS will be designed as a Java EE portal application, and thereby we made the choice for our programming platform.

The programming platform offered us a generic set of building blocks to design a MPS. Designing a MPS based on this platform ensured that the suite could be made available for multiple users who work in a dispersed situation over a mainport organization, who need role based information and interactions, and who need to access distributed data sources. I.e. the building blocks offered by this programming platform could be used to assemble a MPS and deploy it in a standards compliant enterprise information portal.

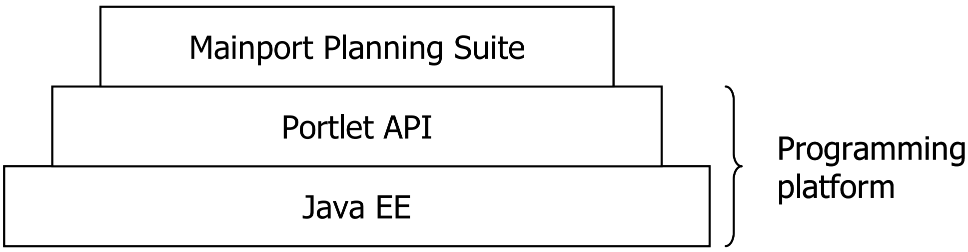


Figure 34: MPS programming platform.

5.3 Identification of the building blocks for a suite to support mainport planning

We chose a programming platform that offered us a generic set of building blocks to design a MPS. In addition to these generic building blocks, we could identify the building blocks needed to support specific requirements related to mainport planning. In chapter 4 we combined knowledge and information visualization to create a framework to extract the functional requirements of the suite. Following the theory of information visualization we used this framework to find promising technologies for accessing raw data, transforming this data, mapping it onto visual structures eventually to create interactive views. To address requirement 4, we mapped the functionalities provided by these technologies into a functional architecture consisting of functional components and their interrelations as is shown in Figure 35. Below we provide an overview of the functional components.

- **Indexer**: An indexer is used to access and index raw data available in the organization.
- **Filter and Blog**: The indexed data is filtered using virtual folders, dynamic visual queries and possible other means. A blogging component provides the meta-data required to effectively filter the raw data.

- **Model**: Visual structures are mapped onto the filtered data using a model. The model depends on the type of data e.g. a GIS model can be used to map cartographic symbols to location based information. I.e. a model provides the business logic for visualization.
- **Visualizer**: A visualizer actually paints the visual structures on a canvas (panel) which is part of a user interface to create the complete image.
- **Portal**: An enterprise information portal aggregates different sources of content into personalized, interactive user interfaces depending on the role and system rights of its users.
- **Service**: The user interfaces are presented as portal services (portlets) which support analysis, design, evaluation and choice in mainport planning.

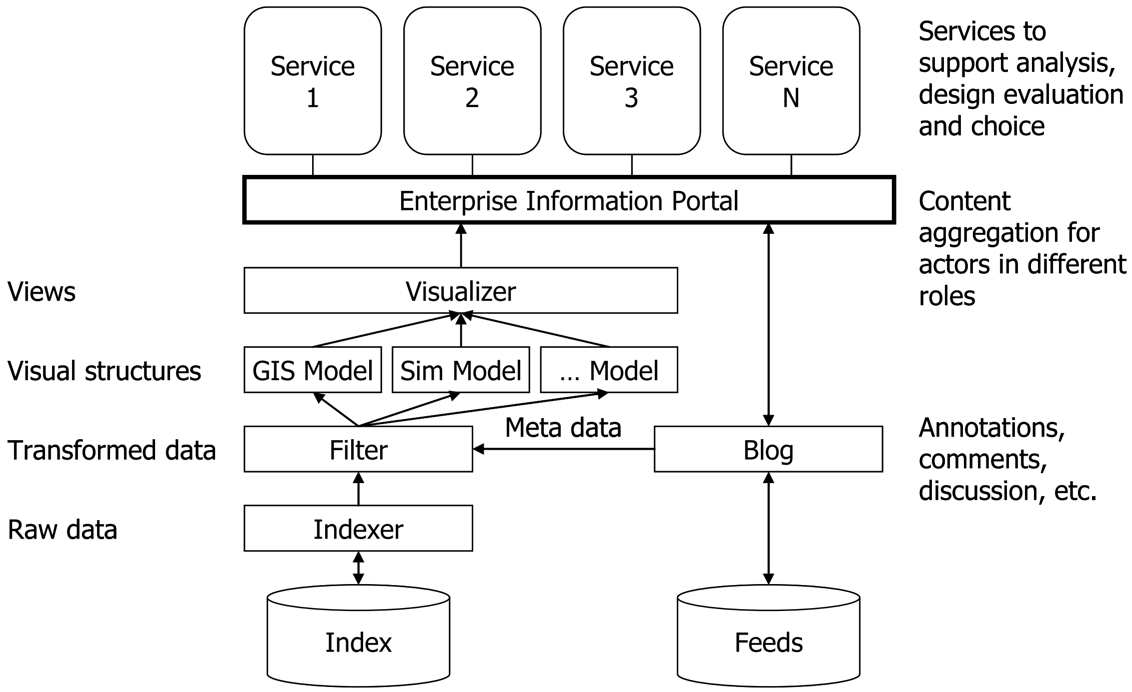


Figure 35: Functional architecture of the Mainport Planning Suite.

We divided each of the identified functional components into a number of loosely coupled subsystems (Figure 36). These subsystems contain the building blocks to assemble the MPS services. The division into subsystems was partly based on the availability of state-of-the-art off-the-shelf code libraries, which fill in a part of the required functionality. Our choice for this specific set of off-the-shelf libraries was influenced by their reusability, availability, quality of coding, and documentation. Therefore we resorted to recognized open-source community projects for the selection of off-the-shelf libraries. These libraries are colored gray in Figure 36. The white subsystems are software libraries that we designed to fill in the functionalities for which no ready to use solutions were available off-the-shelf.

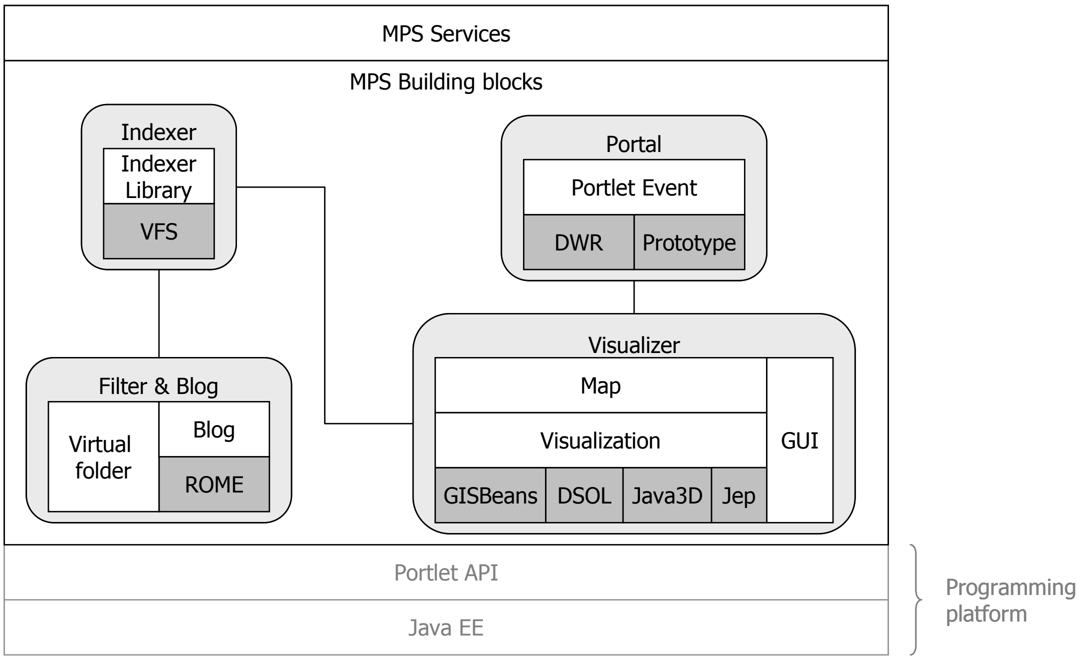


Figure 36: the building blocks for a MPS.

We identified the following subsystems.

Off the shelf libraries

- DWR: Direct Web Remoting, making Java server side objects available in Javascript
- Prototype: a JavaScript library which adds AJAX functionality, Object Oriented functionalities and some utility functions
- Apache Commons Virtual File System, VFS: a library of classes to access a variety of remote and local file systems using URIs such as samba server, ftp servers, zip files and the local file system
- ROME: a blogging library which can produce Atom and RSS feeds
- GISBeans: an implementation of the OpenGIS standards for geovisualizations
- DSOL: the Distributed Simulation Object library
- Java3D: for 3D visualizations and mathematics
- Java Expression Parser, Jep: a library wich provides functionality for parsing math expressions.

For a detailed list of the off the shelf libraries see appendix A5.

Designed libraries

- Portlet Event library: an asynchronous event library that supports both server and client side MVC framework implementations (described in 5.6)
- Virtual folder: a library providing virtual folder functionality (see appendix A1)
- Indexer library: functionality for indexing information from distributed resources (see appendix A2)

- Blog library: reading and writing blog feeds to a database (see appendix A3)
- Visualization library: a library of visualization and animation services (described in 5.4)
- Map library: a library to map visual structures onto GIS data, i.e. create geographic maps, also supporting dynamic geovisualizations (described in 5.5)
- GUI library: Java Swing user interface components such as a “timeslider”, a component which allows the user to select a time interval (see appendix A4)

With the choice for a programming platform and the partitioning of our suite into loosely coupled subsystems based on proven and standardized IT building blocks we addressed requirement 10. In the following sections we introduce the detailed design of the visualization library, geovisualization library, and porlet event library.

5.4 The visualization library

The visualization of temporal and location based information was one of the requirements for our suite (requirement 2). Furthermore, the suite should support geovisualizations (requirement 3). In this section we describe how we support location based information in general. In the following sections we focus on the visualization of geoinformation (i.e. maps) and the visualization of temporal information.

5.4.1 Visualizing location based information

Location based information represents real-world objects such as roads, vehicles, buildings and facilities. This information can be displayed in a two, or three dimensional visualization. Considering the nature of mainport planning and our findings during the case studies, our focus was on 2D visualizations, while we made extensive use of concepts from 3D visualization.

In contrast to 3D visualization, 2D visualizations are essentially layered, just like analogue animation films are made using layers of transparent slides which are stacked to create the complete frame (Pocock and Rosebush 2001). Consequently 2D rendering is substantially less complicated that 3D rendering. Furthermore objects in both a 2D and 3D visualization should be positioned correctly. Positions are defined relative to an axis system. The axis system itself also can have a position, such that all positions are actually relative. For practical reasons there is always one axis system that functions as the base: an absolute zero-position relative to which all other positions can be defined. For example, we can define the location of all cars relative to the center of Paris, while we define the position of the wheels of a car relative to the center of the car.

A scenegraph (Bouvier 2000, Reiners 2002, Strauss and Carey 1992) is a commonly used technique in 3D computer graphics to define the position of objects relative to each other. For example Java3D provides an implementation of a scenegraph (Figure 37). The scenegraph adds an extra layer of abstraction such that the underlying 3D rendering pipeline remains hidden, i.e. the developer is no longer concerned with the technical details of projecting 3D shapes on a 2D screen.

In a scenegraph a location in a “virtual universe” can be described as a collection of objects which are positioned relative to each other. Nodes displayed in the image are:

- BranchGroup (BG) a collection of nodes in the graph
- TransformGroup (TG) a coordinate transformation representing a relative position , i.e. translation, rotation, or scale
- Shape (S): an object to be displayed

Instead of a shape as an end-node, one can also define a viewing platform, which represents the camera position used to render the image.

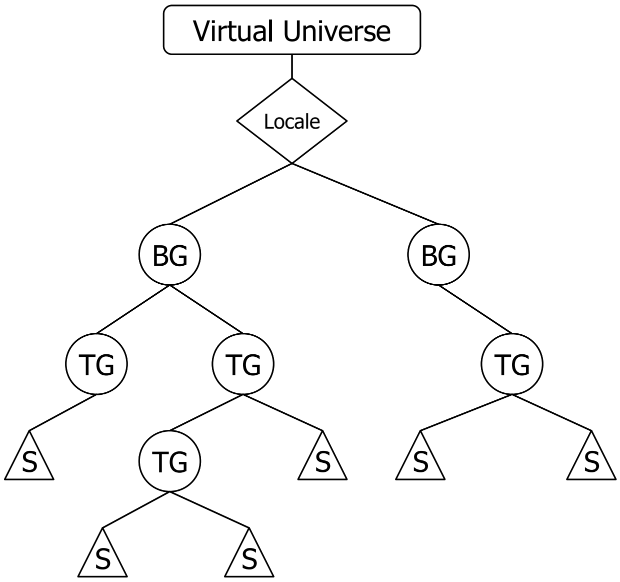


Figure 37: A scenegraph.

Although the concept of a scenegraph is widely supported in 3D computer graphics, 2D visualizations are often constructed using plain drawing functionalities using absolute coordinates e.g. using Java2D (Knudsen and Knudsen 1999). In mainport planning, or spatial planning in general, the use of 2D visualizations is still preferred. Therefore we decided to implement a 2D-scenegraph as a main component for visualizing location based information. The relation between the classes in the 2D visualization library is shown in Figure 38.

- Canvas2D: a panel on which the rendered image is displayed
- Scene2D: the image to be displayed, consisting of 0 to N layers
- Visualizer2D: renders a scene graph; actually represents a layer and therefore implements LayerInterface
- Locale2D: the root of the scenegraph, consisting of 0 to M renderables
- Renderable2D: a (part of) a scenegraph representing a real world object
- BranchGroup (BG): a branchgroup
- TransformGroup (TG): a transformgroup for 2D coordinate transformations
- Shape2D (S) : a 2D shape (e.g. rectangle, circle, polygon etc)

By implementing this 2D-scenegraph we essentially made 2D and 3D visualizations compatible in structure. In 3D we have a *Canvas3D*, *Scene3D*, *Visualizer3D* and so on. The major difference between the 2D scenegraph and a 3D scenegraph is the layered structure

(*LayerInterface*), while in 3D there are no layers. The *Renderable2D* is in fact nothing more than a branchgroup that contains a number of child nodes. Therefore *Renderable2D* is not essential for defining an image, however, each *Renderable2D* should represent a real world system, e.g. a car, that in its turn consists of subsystems, e.g. wheels. In other words: within a renderable multiple branchgroups (subsystems) can exist. A *Renderable2D* therefore conveniently defines the system boundaries, such that we can refer to the system more conveniently than referring to a branchgroup.

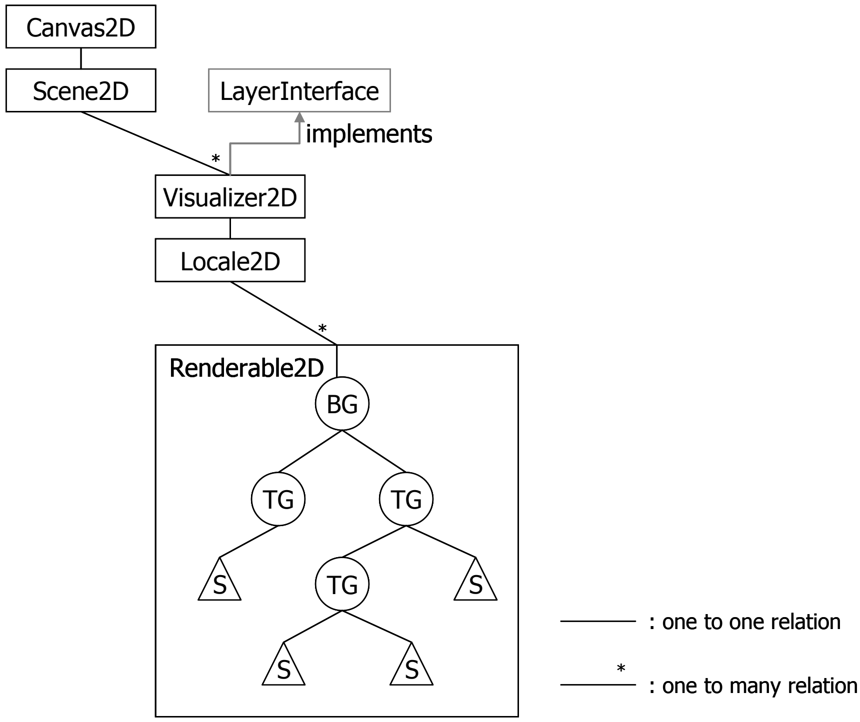


Figure 38: 2D visualization library.

Figure 39 shows an UML class diagram (Fowler and Scott 2000) of the 2D-scenegraph. The structure follows the structure of the 3D-scenegraph offered by Java3D. Node2D and Group2D define the graph's elementary tree structure. Each node has a parent node and groups can contain child nodes. A *BranchGroup* and *TransformGroup* extend Group2D, while a Shape2D extends a Node2D. A *BranchGroup* can be detached from the graph. Furthermore it can contain user data needed for rendering a specific graph. User data can also be used uniquely to tag a *BranchGroup* such that it can be searched for within the graph. A *TransformGroup* has a Java2D *AffineTransform* as a single attribute. An *AffineTransform* represents a coordinate transformation matrix. A Shape2D can actually be painted using a Java2D graphics object. An *ImageObserver* is used in the paint-method for rendering raster images. (An *ImageObserver* enables asynchronous bitmap loading.) In some cases a shape should not be scaled with the rest of the image, therefore it is possible to disable scaling. Furthermore, a computer screen uses an axis system where the x-axis points right and the y-axis points downwards. This can be inconvenient as there are situations where one wants an y-axis that points upwards. We can transform (or "flip") the axis system such that the y-axis points upwards. This also means that e.g. text labels are drawn upside-down, therefore we added the option to neglect axis system flipping for individual shapes.

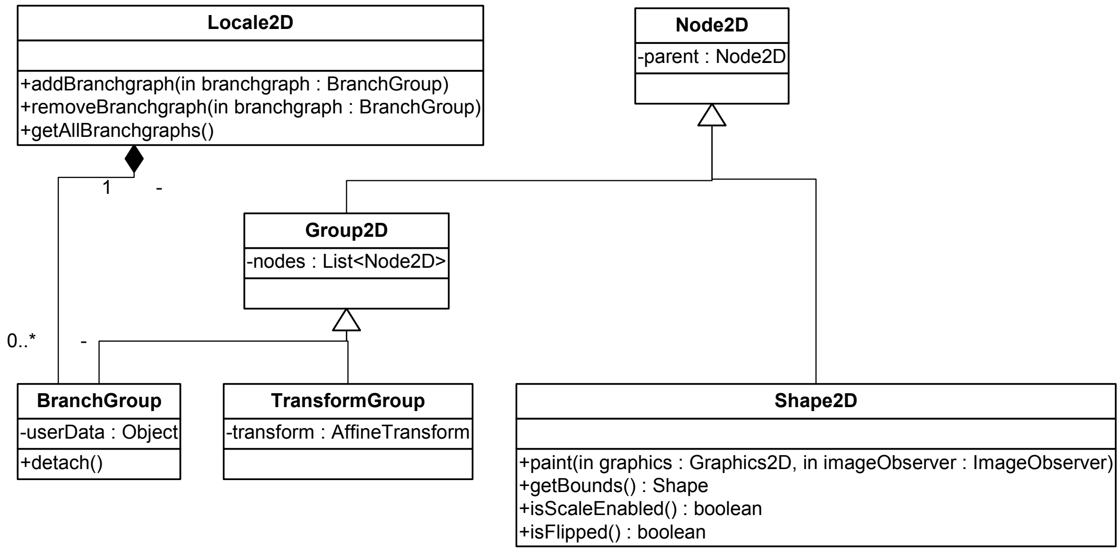


Figure 39: Class diagram of the 2D-scenegraph.

Figure 40 displays an UML class diagram of the classes that are used for the actual rendering of the image. A *RenderableInterface* is an empty interface just meant to identify a renderable, which can be either a 2D renderable in a 2D-scenegraph or a 3D renderable in a 3D-scenegraph. Similarly a *VisualizerInterface* is not specific for 2D or 3D visualization. The specification happens in the actual Visualizer2D that implements a *LayerInterface* and extends a *BranchGroup*. A Visualizer2D recursively transfers through the nodes of the scenegraph. The transformations in *TransformGroups* are concatenated to determine the absolute positions of the shapes. A layer has a name, can be enabled or disabled, and can be painted on a canvas. Consequently it is possible to create a class that implements a *LayerInterface* to draw on a canvas without making use of the scenegraph concept. This might be desirable in situations in which only absolute positions are used. A Canvas2D is a specification of a Java Swing *JPanel* component. It has an extent: a rectangular shape which specifies the position and size of the canvas in world-coordinates (in contrast to pixel coordinates). In other words, the extent defines the viewport through which the image is displayed, and is in that sense equivalent the a viewing platform (camera) in Java3D. However, the extent is not part of the scenegraph while a viewing platform is. Although it is not impossible to implement an extent as part of the scenegraph, we consider our solution to be sufficient and less complicated. Finally a Canvas3D provides methods of converting positions in screen-coordinates (pixels) to and from world-coordinates (distances in e.g. meters).

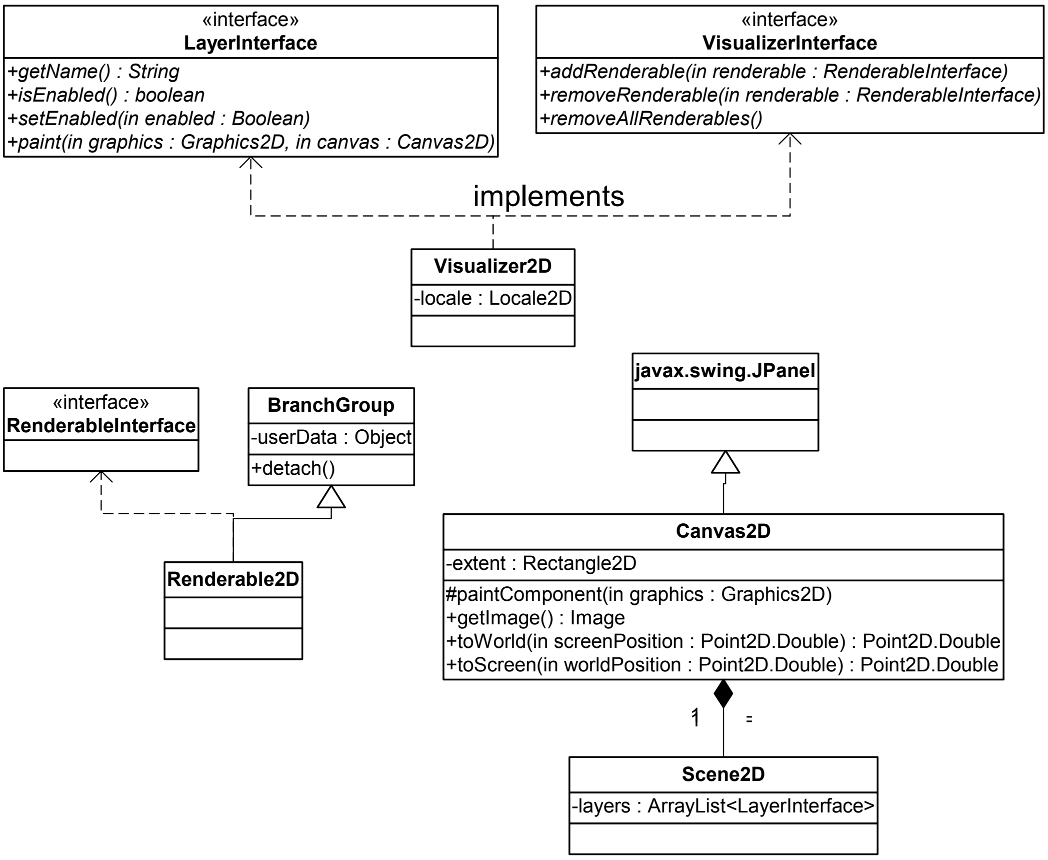


Figure 40: Class diagram of 2D visualization.

5.4.2 Summary of design decisions

In this section we presented the 2D visualization library and thereby we addressed requirement 2 with respect to visualization of location based information. We can now summarize the most important design decisions made for the design of this library.

- we based the structure of the library on the scenegraph concept, because this creates an extra layer of abstraction that hides the complex operations required for transforming relative positions to absolute positions and vice versa
- we based the design of the scenegraph on the Java3D scenegraph, because Java3D provides a well defined, complete and widely used scenegraph
- in contrast to the Java3D scenegraph, the 2D scenegraph does not support a z-coordinate, instead we used layers to define the order in which overlapping images are painted
- we introduced a *Renderable* as an empty interface to tag a *BranchGroup* as a system consisting of subsystems, such that it can easily be referenced in the scenegraph
- we introduced a 2D canvas that converts an image in world-coordinates to screen-coordinates, such that the developer can simply use real world units (e.g. meters) without bothering about how this would fit on the screen

5.5 The map library

Now that we have designed a library for displaying location based information, we still need functionality to create geovisualizations i.e. maps (requirement 3). Geographic information represents a model that needs to be visualized using the visualization library. Following the theory of information visualization as described in chapter 3, the map library is used to map geographic information to visual structures. For example, it can be used to map contours onto noise levels, or color coding to specific industry types, or circular markers to facilities in a port, e.t.c..

In practice this means that we needed functionalities to read and write map specifications, vector and raster images. Furthermore we needed functionalities to create scale bars, labels and legends (Slocum et al. 2005). We first focus on a design to support static geovisualizations. after which we extend our design to support dynamic geovisualizations.

5.5.1 Symbolizing GIS data

GIS software is commonly used to create static geovisualizations (maps). The principles of symbolization are used to map visual structures onto GIS data. Apart from commercial software such as ArcGIS[19], there are also open source GIS packages. To integrate GIS functionalities in our suite we searched for open-source, and preferably Java-based GIS libraries. Below we describe the library that we found and we motivate our choice for a preferred library.

MapServer

MapServer (http://mapserver.gis.umn.edu/) is an open-source development library for building "spatially enabled internet applications". MapServer essentially provides a server-based CGI application that can be queried over the Internet to produce maps that can be displayed in a webpage. The map to display is specified in a map-file, which is a text-file that describes how the map should be generated and which resources should be used. MapServer, however, is not written in Java but is claimed to support scripting in Java. Although MapServer is an advanced GIS library, we decided not to make use of MapServer for the following reasons.

- MapServer is not written in Java, which makes it difficult to reuse its components in our suite.
- MapServer does pure server based rendering, which does not conform to the concept of rich clients as described in chapter 4.
- MapServer produces static maps, while we also want to display temporal information.

GeoTools

GeoTools (geotools.codehouse.org) is a pure Java based GIS library that implements the OpenGIS standards. Therefore GeoTools seemed to be a viable option for creating geovisualizations in our suite. However, we decided not to make use of Geotools for the following reasons.

- At the moment of writing, Geotools is under active development and therefore the API is rapidly changing, resulting in inconsistent and incomplete documentation.
- GeoTools depends on a large number of other software libraries and the library itself is split up over a large number of jar-files[20]. We could not find proper documentation on

[19] ArcGIS is a GIS product of ESRI.
[20] A jar-file is a Java archive file, which is used to store compiled programming code and other resources.

the dependencies and interrelationships of these libraries, which would significantly complicate the use of GeoTools.

Thus although GeoTools definitely has potential as a standard compliant GIS library, we did not make use of it for practical reasons.

Google Maps

Google Maps (maps.google.com) is a geovisualization service introduced by Google in early 2005. A developer API was introduced in June 2006[21]. It offers an AJAX-based client to visualize a fully interactive world map that is rendered on a server. Interactive markers and overlays can be added to the world map to indicate specific locations such as shops, hotels and other businesses. Google Maps seems to be a promising solution, but we did not use Google Maps for the following reason.

- While Google Maps was introduced in 2005, the developer API was introduced after we finalized our design, and it is only available for developing solutions for public websites. Therefore we could not make use of Google Maps.

GISBeans

GISBeans (http://sourceforge.net/projects/gisbeans) is a light-weight Java-based GIS library. It provides functionalities for reading ESRI shape files. Furthermore it provides a parser for XML-based map specifications. Although not as extensive as MapServer and Geotools, and although not complying with the OpenGIS standards, we chose to use GISBeans for the following reasons.

- GISBeans provides a relatively simple structure with well written and documented source code. Therefore we can easily decide which parts of Geotools we can reuse and integrate into our suite.
- GISBeans is already used in the DSOL simulation suite (in fact it was developed by the same researcher).

We therefore chose to use the GISBeans library to provide GIS functionality in our suite. However, we did not use the complete library. GISBeans provides a class called *Map* which implements an interface called *MapInterface*. *Map* represents a model of a map consisting of layers, which in turn consist of shapes. In addition *Map* provides methods for painting the map, using its own extent for coordinate transformations. Directly making use of the Map-class would break compatibility with our 2D-visualization library. Instead we could make use of the functionalities provided for reading in ESRI shape files. Furthermore we could reuse parts of the code for parsing xml-based map-files.

The MapServer and Geotools projects, in compliance with the OpenGIS standard, make a distinction between the data to display, and the style in which these data are displayed. For example, the data can represent noise-contours where every level of noise can be displayed in a different color (style). The xml-based map-file specification of GISBeans, however, does not have this distinction: data and style are completely intertwined. Consequently, applying a different style to available data becomes a tedious task. Therefore we decided to redefine the xml-based map-file such that it follows the same structure as MapServer's map-file (see appendix B).

[21] See http://googlemapsapi.blogspot.com/2006/06/geocoding-at-last.html

We recognize that our approach could be tricky, in the sense that mainly for practical reasons we choose GISBeans, while Geotools may eventually get to a stage that makes it more attractive than GISBeans. Furthermore, the xml-based map-file specification does not follow a commonly accepted standard. To compensate for these potentially tricky choices we made extensive use of the factory pattern (Gamma et al. 1994) in our code. The factory pattern is a software development pattern that decouples interfaces from their actual implementing classes. The advantage is that someone can replace our xml-based map-file parser with a parser for another file type without breaking the rest of the code.

For example, one would normally write:
```
MapInterface map = new Map(uri);
```

Where using a factory pattern one can write:
```
MapInterface map = new MapFactory().create(uri);
```

The *MapFactory* returns a map. When using a factory pattern it is not necessary to know which class is used to implement the *MapInterface*. When, at a later stage, another implementation of *MapInterface* becomes available, the *MapFactory* can be modified to produce (instantiate) the new implementation of *MapInterface*.

A conceptual UML class diagram (Larman 2004) of the map library and its relation to the visualization library is shown in Figure 41. The map library represents a logical model of a map, while the visualization library is used to generate a visual model. The Map class represents the map to be displayed. It consists of Layers that extend Visualizer2D. Each Layer contains data. Data represents the data that is actually used. Data can reference multiple sources such as ESRI shape files and raster images. In compliance with the map-file structure of MapServer each Layer can have multiple *LayerClasses*. A *LayerClass* defines a style for the data and how a label should be displayed, font type, color etc.. Multiple *LayerClasses* can be defined to represent different types of data, e.g. representing water and land.

Based on the *Data* a *Layer* instantiates a *GISImage*, which is the actual shape (*Shape2D*) to be displayed. One can think of a *GISImage* as the result of processing *Data* and *Style*. The *Layer* holds a soft reference to the *GISImage* to prevent a memory overflow in the Java Virtual Machine
(see http://java.sun.com/j2se/1.4.2/docs/api/java/lang/ref/SoftReference.html).

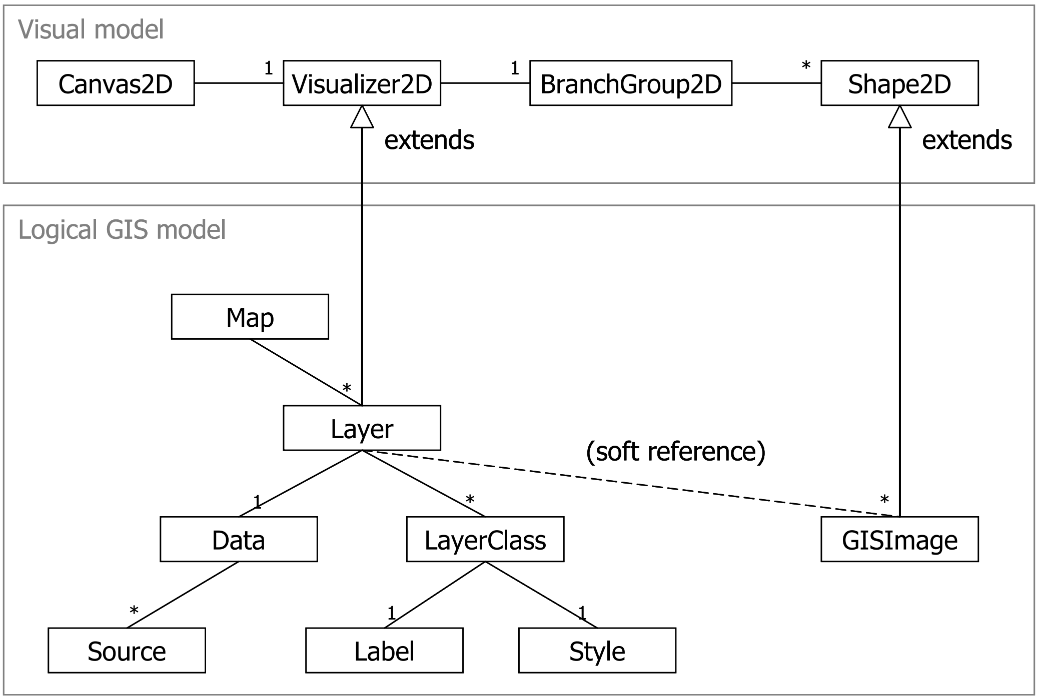


Figure 41: Conceptual UML class diagram of the map library.

A *GISImage* can in principle be of many types. It can represent an ESRI shape file, a TIFF raster image, a JPG-file, or something else. It was not possible to know in advance what types of data our library would need to provide support for during its lifespan. For now we chose to support ESRI shape files through the functionality offered by GISBeans and JPG-images through the standard Java API, but tomorrow we might need to support another type of data. Therefore we again resorted to the factory pattern as is displayed in the UML class diagram in Figure 42.

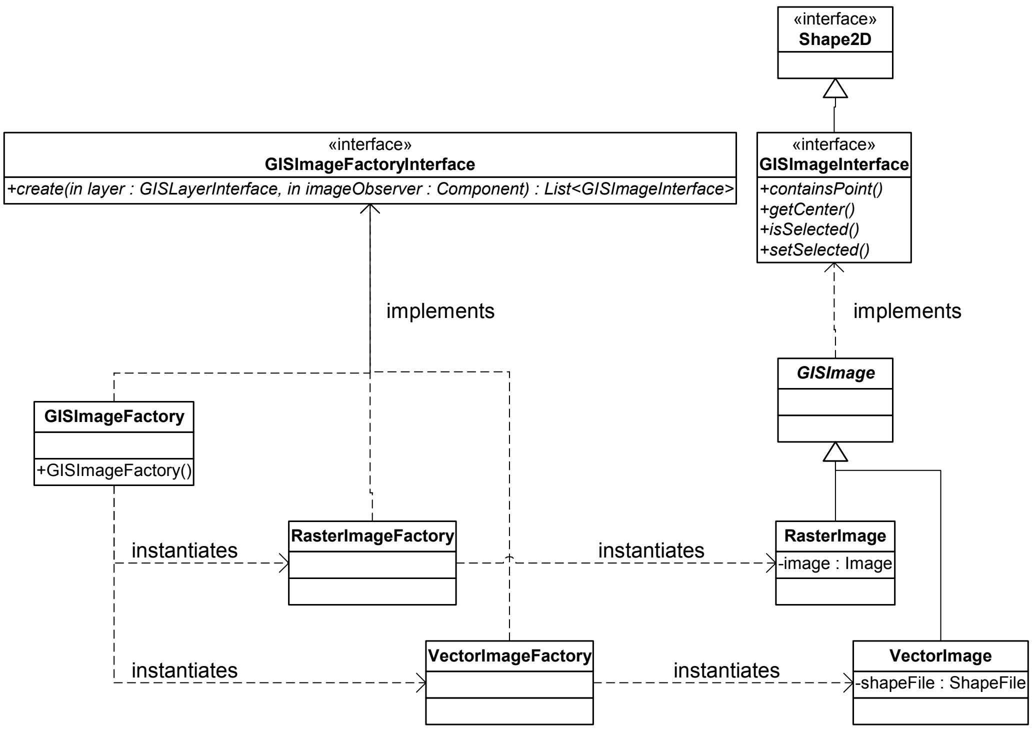


Figure 42: Using a factory-pattern to instantiate GIS images.

5.5.2 Animating temporal GIS data

We have described how the suite supports location based information and how geovisualizations are supported, but we have not yet addressed the visualization of temporal data (requirement 2). To support the animation of temporal GIS data, such as changes of land use over time, we defined a *DynamicGISLayerInterface*. This interface extends the functionality of a static GIS layer defined in *GISLayerInterface* by implementing an *EventListener* - interface. The *EventListener* is specified in the DSOL simulation library to support asynchronous events among objects. Furthermore a *DynamicGISLayer* has an *Interval,* which represents a time interval wherein the layer is active (read visible). The layer listens to events that specify a change in clock-time. The clock-time is the actual moment in time to be displayed, or an interval of time to be displayed. The clock-time is compared with the specified interval to determine if the layer should be displayed. Similarly, it is also possible to activate or deactivate individual GIS images depending on the clock-time, through the definion of a *DynamicGISImageInterface* which extends *GISImageInterface*, see Figure 43 below.

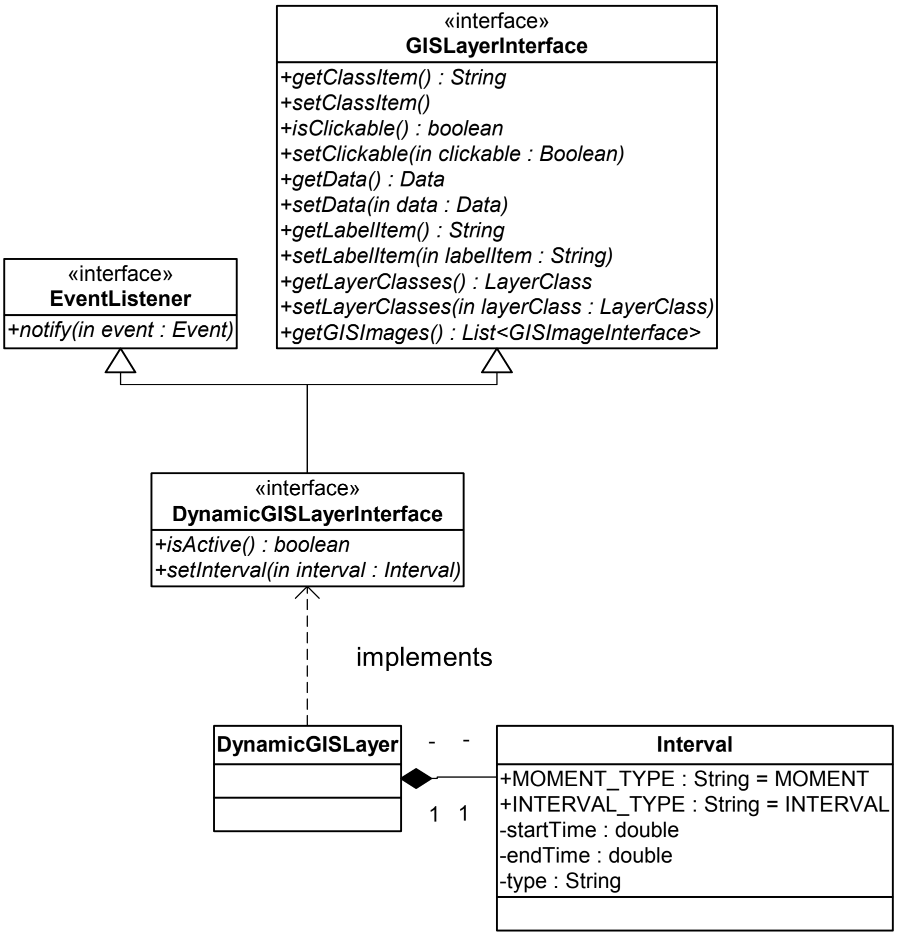


Figure 43: Dynamic GIS layer.

5.5.3 Summary of design decisions

We introduced the design of the map library and thereby we addressed requirement 3. Furthermore we addressed requirement 2 in section 5.5.2 with respect to the visualization of temporal information. The important design decisions made for producing the map library are summarized below.

- We decided to reuse some of the functionality available in the off-the-shelf GISBeans library, however we decided not to reuse the entire GISBeans library because it does not decouple data, visual structures and the actual visualization of maps
- We decided to base the map structure on the map-file definition of MapServer because it does provide a clear separation of concerns between visual structures (style) and data
- We made extensive use of the factory pattern to create flexibility and extensibility in our design
- The animation of temporal GIS information is achieved through asynchronous events, which effectively decouples the map library from the source of the change in clock time

5.6 The Portlet Event library

In this section we address the usability requirements, mainly focusing on the requirement for rich clients in web portals (requirements 7, 8 and 9). Recently two new technologies have been introduced in the design of web applications: the portlet specification (JSR168) and Asynchronous JavaScript And XML (AJAX). As explained in chapter 4 we wanted to make use of these technologies in our MPS. However, the technologies are incompatible: the portlet standard supports the classic server-based MVC framework, while AJAX moves the MVC framework to the client.

Our Portlet Event library was designed to overcome this incompatibility. We should note that the open-source community also recognized the limitations of JSR168 with respect to rich clients (Ziebold and Sum 2006). Therefore a new standard is under development at the moment of writing: JSR286, Portlet Specification 2.0 (Hepper 2006). JSR286 is still in draft and no reference implementation was available at the time of writing (2007). We will compare our solution to JSR286 at the end of this section.

5.6.1 Supporting rich clients in the Mainport Planning Suite

The rich client architecture that we achieved is shown in Figure 44. On the server side the suite should access (distributed) data sources in the mainport organization. Libraries of software components (models) are used to process this data. Next portlets and servlets control how the information is displayed in a user interface and how the user interaction is handled. Java Server Pages produce the user interface. Finally JavaScript is used for client-side user interaction and AJAX requests. In addition it is possible to embed applets, and Java Web Start applications, in the user interface. Ideally applets communicate with the server-side code through the JavaScript layer. Consequently multiple MVC workflows are achieved throughout the suite. Portlets and servlets handle user interaction and control the business logic on the server, while JavaScript is used in HTML-based user interfaces on the client.

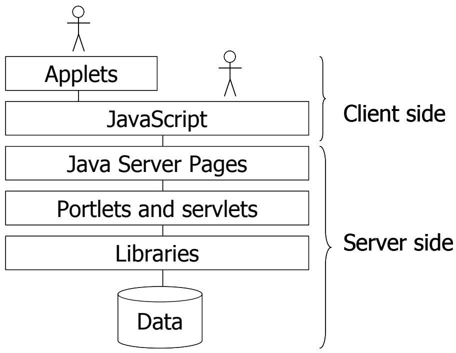


Figure 44: Rich client architecture.

5.6.2 Asynchronous event handling

Portlets handle user interaction on the server and thereby take the role of a controller in the suite. As we described in chapter 4 the controller dispatches user requests to handlers to execute the business logic. Portlets however, following the portlet specification, handle user interaction as shown in Figure 45. First the *doView* method generates the portlet's user interface, commonly by dispatching the request to a Java Server Page (JSP). Next the user interacts with the user interface e.g. by clicking a button. The button sends an HTTP request back to the server. This request is handled in the *processAction* method. After handling the request the portlet can respond in two ways: (1) call the *doView* method to update the user interface, or (2) dispatch the response to an external URL, e.g. another website.

A JSR168 compliant portal thus follows a classic MVC framework, since all interaction is handled on the server and the server responds by sending an updated user interface after each request. However, a portlet does not necessarily dispatch user requests to handlers as *processAction* can in principle take the role of a handler and can even contain the business logic, i.e. the standard does not specify how portlets can or should be used in a true MVC framework.

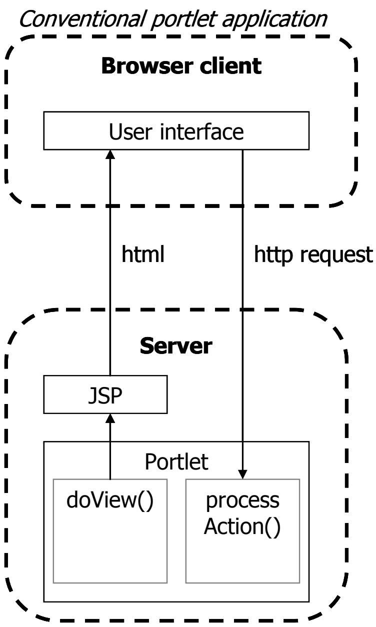


Figure 45: The workings of a JSR168 compliant portlet application.

Truly rich clients should also support asynchronous event handling because it decouples the provided functionalities. On the server-side, the *processAction* method of a portlet does not provide this. On the client-side JavaScript should be used to provide asynchronous event handling.
The observer-pattern (Gamma et al. 1994), also known as the publish-subscribe interaction scheme, provides a loosely coupled form of interaction between the various subsystems of IT systems (Figure 46). We can distinguish a publisher, an event service and a subscriber, effectively providing a one-to-many relationship between a publisher and a subscriber. Messages from a publisher to subscribers are called events and subscribers are notified whenever an event arrives. Eugster et al. argue that three types of decoupling can be realized with this scheme:

- space decoupling: publishers do not usually hold references to subscribers, and subscribers do not usually hold references to publishers
- time decoupling: the interacting parties do not necessarily participate in the interaction at the same time, events can be buffered
- synchronization decoupling: the production and consumption of events does not interrupt the flow of control in either publisher or subscriber

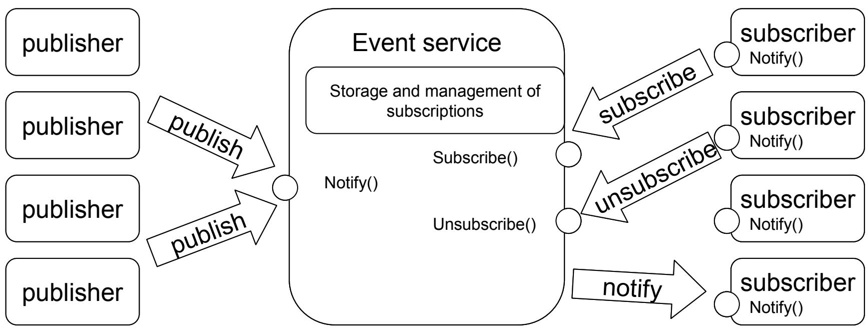


Figure 46: An example of a publish-subscribe system. (source: Eugster 2003)

In the design of an MPS especially we see the advantage of space decoupling for handling user interaction. When a user clicks a button, edits a text field, or moves a slider bar this produces events which should be processed by the business logic. When the suite grows in size and complexity space decoupling increases maintainability and reusability by decoupling functionalities.

5.6.3 Asynchronous event handling in portlet applications

Different solutions exist to establish asynchronous event handling in web applications such as Apache Struts[22] and Java Server Faces, JSF (part of Java EE). However these solutions support servlet applications and not portlet applications because they were introduced before the portlet specification. Since the introduction of the portlet specification in 2004 several portal vendors such as IBM (WebSphere portal[23]) have developed “Struts” for portal applications. As far as we understand these solutions create a lock-in on vendor specific portals, while one of the main drivers for the portlet specification was to establish a cross-

[22] See http://struts.apache.org/
[23] See http://www-306.ibm.com/software/websphere/

portal standard. Although the open-source community[24] is working on cross-portal MVC framework solutions these were not available when we started our design.

Therefore we designed a lightweight MVC framework for portlet applications which offers asynchronous event handling. We argue that using this lightweight MVC framework enables the fulfillment of requirement 9. When cross-portal MVC Frameworks such as Struts become widely available our framework can be replaced by these more advanced, but also more complicated, solutions.

Our server-side MVC framework makes use of actionURLs on the client to create user events. These are parsed and processed on the server. We shall now explain the details of this solution. As we explained in the previous section, portlet applications commonly use JSP to generate the view, the user interface (Figure 47, A). The JSP page produces the HTML which represents the user interface. When the user presses a hyperlink or submit-button a HTTP action request is send back to the server. JSR168 provides a set of JSP-tags to construct an action request e.g.:

```
<portlet:actionURL var="actionlink">
      <portlet:param name="param1" value="some value 1" />
      <portlet:param name="param2" value="some value 2" />
      …
      <portlet:param name="paramN" value="some value N" />
</portlet:actionURL>
<a href="<%=actionlink%>">Send action request</a>
```

The action request is parsed into the URL of the portal page. The portal sends the provided parameters to the *processAction* method of the originating portlet, which then handles the request. Finally the portal calls the doView method to update the user interface.

We extended the *GenericPortlet* to implement asynchronous event handling following a publish-subscribe pattern (Figure 47, B). The portlet becomes an event producer which delegates user interaction requests to event listeners. The result is a space decoupling of events which are handled on the server.

[24] See http://portlets.blogspot.com/ (an entry point for state-of-the-art portlet technology, which is widely recognized in the open-source community)

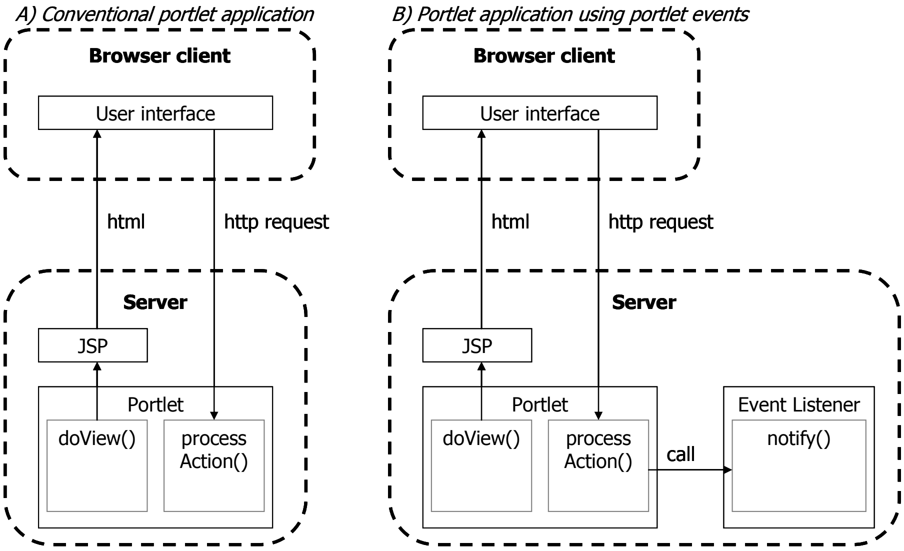


Figure 47: A classic portlet application (A) versus a portlet application using portlet events (B).

Figure 48 shows an UML class diagram of the specification of portlet events. A *PortletEvent* holds a reference to the source of the event. Usually this is the originating portlet, or alternatively the name of the originating JSP page, a portlet can make use of multiple JSP pages. Note that the source is referenced as a string not an *java.lang.Object*, because the client-side has no knowledge of the corresponding portlet object instance. Similarly the event type and value are strings. Furthermore the *PortletEvent* wraps the *actionRequest* and *actionResponse* parameters to access the portal. The event is fired by a *PortletEventProducer* which notifies all registered listeners, which implement *PortletEventListenerInterface*.

To call the corresponding event listener object(s) we specify an action request as follows:

```
<portlet:actionURL var="actionlink">
      <portlet:param name="source" value="aPortlet" />
      <portlet:param name="eventType" value="SOME_EVENT" />
      <portlet:param name="value" value="some value" />
</portlet:actionURL>
<a href="<%=actionlink%>">Send request</a>
```

Where:

- source names the originating portlet
- eventType is the event type used
- value is the value sent

Source, eventType and value are action request parameters and can thus be accessed through the actionRequest object. The value parameter is in that sense not required, but convenient for situations where only one value has to be sent to the server.

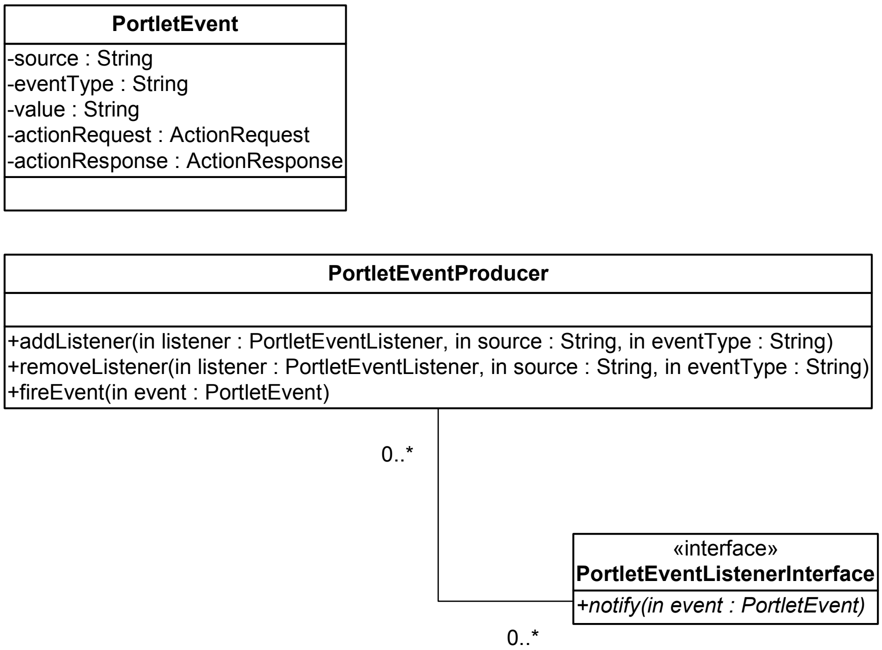


Figure 48: Class diagram of the portlet event mechanism

5.6.4 Handling AJAX events in portlet applications

Despite the space decoupling achieved, events are still handled on the server in the classical way which prevents the implementation of truly rich clients in a portlet application. Following Asynchronous JavaScript And XML (AJAX) we could extend our solution to support the development of rich browser-based clients (Figure 49). User interaction results in a JavaScript call to an AJAX Engine, i.e. JavaScript code, which produces an asynchronous resource request to the *processAction* method of the corresponding portlet. Usually after handling a request in *processAction* the portlet responds by calling the *doView* method to send an updated user interface to the client. However, following the AJAX-paradigm only data should be sent back to the client, not an entire user interface. Therefore the AJAX request should be distinguished from a conventional http request. The portlet can distinguish the request from a conventional http request by including an extra parameter in the request e.g. actionType=RESOURCE e.g.:

```
<portlet:actionURL var="resourcelink">
      <portlet:param name="source" value="aPortlet" />
      <portlet:param name="eventType" value="GET_TIME_EVENT" />
      <portlet:param name="value" value="GMT+1" />
      <portlet:param name="actionType" value="RESOURCE" />
</portlet:actionURL>
<a href="#" onclick="loadContent('<%=resourcelink%>')">Send request</a>
```

The portlet still handles events using event listeners, but the response is encapsulated in a redirect request to a *ResourceServlet*. The *ResourceServlet* then sends back the response to the AJAX client. This response can be a plain text message or an XML document which is parsed on the client.

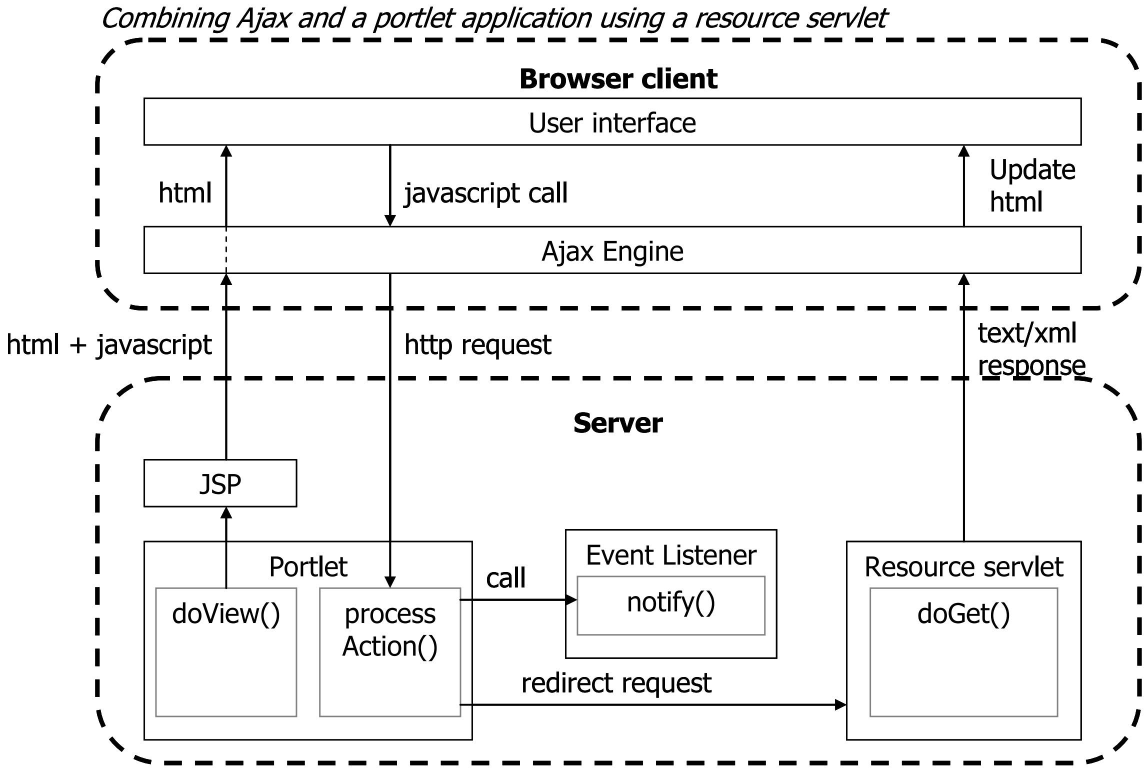


Figure 49: Combining AJAX and a portlet application using a resource servlet.

Ziebold and Sum (2006) propose another solution. Their solution is to call a resource servlet directly from an AJAX client. We could extend their solution by designing an event mechanism similar to portlet events as introduced in section 5.6.3. The disadvantage of this solution is that a direct call to a servlet bypasses the services offered by a portlet such as user authentication and access to portlet attributes. Ziebold and Sum (2006) propose a workaround to solve this problem. This workaround, in a sense, accesses a portlet's services through the "back door". We refer to their article for those interested in the details of this workaround.

As another possible alternative is Direct Web Remoting (DWR) (Figure 50). The off-the-shelf DWR library dynamically creates JavaScript objects which have an equivalent Java object on the server. Consequently JavaScript programs can use server-side objects as if these were available on the client. This results in an elegant solution that enables asynchronous event handling on the client. However, it adds complexity on the server as object instances which are made available through DWR should be registered in advance in a dwr.xml - configuration file. This complexity is not always desired nor required. Furthermore it suffers from the same problem as Ziebold and Sum's solution: it bypasses the services offered by a portlet. This can be solved using a workaround similar to that proposed by Ziebold and Sum (2006).

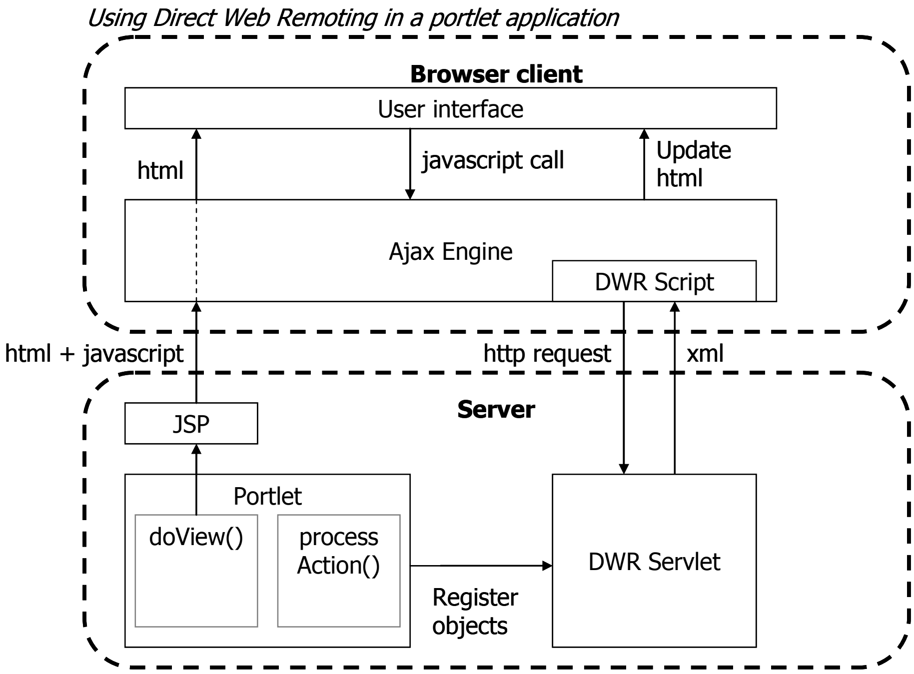


Figure 50: Using DWR in a portlet application.

5.6.5 Client side event handling

Event handling is also required on the client side when (1) the user action does not produce a server request, or (2) when the server request is handled via AJAX. Applets and Java Web Start applications can use the standard Java Swing action, change, and mouse listeners. Browser based user interfaces make use of Javascript for handling user events. Javascript in itself does not provide functionality for event handling following a publish subscribe interaction scheme. However, the open source Prototype library provides a useful asynchronous event handling mechanism. The following JavaScript class illustrates the use of event handling on the client side. Note also that we made use of Prototype's Class function, which makes the essentially procedural JavaScript language feel more like a truly Object Oriented language. Furthermore we used Prototype's $-function as a shortcut to reference HTML elements from the webpage, note $ is a function name!

```
/**
 * Class FileActions
 * File actions
 * depends on Prototype
 */
FileActions = Class.create();
FileActions.prototype = {
      /**
       * Constructor.
       */
      initialize: function() {
            // Create our buttons
            Event.observe($('removeFileButton'), 'click',
               this.onRemoveFileButton.bindAsEventListener(this), false);
            Event.observe($('newFileButton'), 'click',
               this.onNewFileButton.bindAsEventListener(this), false);
            Event.observe($('deleteFileButton'), 'click',
               this.onDeleteFileButton.bindAsEventListener(this), false);
```

```
            Event.observe($('editFileButton'), 'click',
               this.onEditFileButton.bindAsEventListener(this), false);
      },

      /** Called when pressing the remove file - button */
      onRemoveFileButton: function(event) {
            // Removes a file ...
      },
      /** Called when pressing the new file – button */
      onNewFileButton: function(event) {
            // Creates a new file ...
      },
      /** Called then pressing the edit file – button */
      onEditFileButton: function(event) {
            // Edits a file ...
      }
}
```

5.6.6 Summary of design decisions

We introduced the design of the portlet event library and thereby we addressed requirements 7, 8 and 9. Design decisions made regarding the portlet event and AJAX library are listed below.

- We based our design on JSR168 portlet standard and AJAX. JSR168 is a widely supported industry standard for portlet applications. Following this standard we prevent a lock in on a specific portal vendor. Consequently the lifespan of our design is not limited to that of a specific portal product. AJAX is still a relatively new software development paradigm which has not yet found (2007) its way to common portlet applications.
- We decided to decouple the different functionalities through asynchronous event handling using the observer pattern.
- We decided to provide asynchronous event handling both on the server and on the client.
- We introduced different methods for asynchronous event handling on the server: through a portlet, using a servlet or using DWR.
- Client side event handling can be provided through the off the shelf Prototype library.

As we promised in the beginning of this section (5.6), we will now compare our solution with JSR268, the new portlet specification which is currently being developed (Hepper 2006). JSR268 is an extension of JSR168: it offers the same interfaces but also adds functionality for AJAX compatibility and communication across portlet applications. To support resource requests it offers the *ResourceServingPortlet* - interface, which processes an AJAX request and sends back content (not a webpage). This is a great improvement which overcomes the incompatibility of the Portlet Specification with AJAX clients. It ensures that an *actionType* - parameter or a workaround as proposed by Ziebold and Sum (2006) is no longer necessary. However the new portlet specification does not offer asynchronous event handling as in our solution. Action requests and resource requests are still handled in a straight forward manner, i.e. not using action handlers or resource handlers (event listeners). The development of vendor independent Struts frameworks, or JSF will eventually provide cross-portal MVC frameworks.

5.7 Conclusions

In this chapter we introduced the design of the building blocks for our MPS. We first chose Java EE extended with the portlet specification as our programming platform. Based on this platform and the requirements for our suite, we identified a number of functional components which we further divided into a number of subsystems (libraries). A number of libraries were available off-the-shelf, and other libraries had to be designed by us. We specifically focused our design at visualization, geovisualization, and enabling the use of rich clients in our suite.

One important observation is that the software development community is progressing very fast. Often their output channel is the open-source community, i.e. technology websites, forums, wikis, blogs, and mailing lists, and not the scientific community, i.e. via journals, academic conferences. Software vendors carefully watch developments in the open source community, and often open up their software for open source development. Since we started our design in 2004 Google has introduced geovisualization for the masses with the introduction of Google Maps and Google Earth, and in early 2007 Microsoft followed by introducing Virtual Earth. Sun Microsystems decided to make their Java platform open-source to boost development. Furthermore developers quickly ran into the limits of the portlet specification, especially when trying to integrate existing MVC frameworks. Several initiatives have been set to overcome the limitations of the current portlet specification.

Therefore we can conclude that the gap between our requirements and the available state-of-art software is closing quickly. While, this will make part of our designed building blocks obsolete, it also means that IT technology is now ready for designing and implementing studio-based mainport planning systems.

In the next chapter we describe the designed services offered by our MPS, based on the building blocks we introduced in this chapter.

6 A Mainport Planning Suite

In this chapter we introduce the functional design of our Mainport Planning Suite, MPS. We used the technical building blocks, introduced in chapter 5, to design and assemble our suite of mainport planning services. These services specifically address requirements 1 and 5 as stated in chapter 4. We designed the mainport planning services in an iterative design cycle in close cooperation with a team of experts in the Port of Rotterdam, PoR. Consequently, the examples which we used to explain our design in this chapter are based on information provided by the PoR such as geographic maps, and, fictive but realistic, project data. To gain feedback on our design ideas we extensively used prototyping. However, for the sake of clarity, we separated the design of our services from their instantiation in prototypes. In this chapter we focus on the design of the services of our MPS and in chapters 7 and 8 we address the prototype implementations of our MPS that we used to evaluate our design.

We first introduce our view on mainport planning services in section 6.1. In the sections that follow we subsequently introduce the design of the services in our MPS. We made a distinction between two types of services: services which support the content of mainport planning and services which support the process of mainport planning. The services which support the content of mainport planning refer to the activities: analysis, design, evaluation and choice. The services which support the process of mainport planning refer to activities such as setting agenda's and agreeing on actions. Our description focuses mainly on the services that support the content of mainport planning, as these are specifically designed based on the architecture that we described in chapter 5. For each service we describe its purpose, how it supports the planning process and its design, i.e subsystems and design choices.

6.1 Mainport Planning Services

Before introducing the design of the services of our MPS, we will sharpen our definition of services. How do we see mainport planning services? How do services differ from the building blocks which were introduced in chapter 5? In chapters 4 and 5 we stressed the importance of visualization and human computer interaction in the design of our MPS. To address the objective of our suite we have to go one step further. We adhere to Streitz and Nixon (2005) who state that the ultimate goal is not to interact with computers, but to interact with information, to communicate and to collaborate with people. They argue that computer-based support should augment these activities. In addition we adhere to Coutaz et al. (2005) who argue that computer-based support should be context aware. Coutaz et al. state that: "Context is not simply the state of a predefined environment with a fixed set of interaction resources. It's part of a process of interacting with an ever-changing environment composed of reconfigurable, migratory, distributed, and multiscale resources." They state that a process goes through different stages and in each stage the actors involved can take on a different role. Depending on the actors' role and the stage at which they are, the kind of computer-based support they require fluctuates.

We aimed to design loosely coupled services that meet the changing information and visualization needs of mainport planners, depending on their role and activities in mainport planning projects. We adhere to Papazoglou and Georgakopoulos (2003) in that a service provided by an IT system is a self-describing, open component that supports the rapid, low-cost composition of a distributed information system. Chesbrough and Spohrer (2006) state that Web Services and the Service Oriented Architecture (SOA) support the codification of identifiable business processes. Web Services take care of the transfer of information between distributed IT systems. They state that SOA forms the foundation for automated B2B information exchange using Web Services. Endrei et al. (2004) state that a SOA consists of three main elements each of which play a different role: a service provider, a service consumer and a service registry. A Web Service comes with a service description which is published in a service registry, such that it can be found by a service client. After a service is found a service consumer can bind to a service provider to invoke the service. The services that we designed for our suite are not Web Services. While Web Services are designed for automated information exchange between distributed IT systems, the services provided by our MPS are designed for providing visualized information to human beings. We follow the lines set out by Streitz and Nixon (2005) and Coutaz et al. (2005): our services represent the front end of an information system, where the human is in the loop. Although not Web Services, the services provided by our MPS are built on an architecture that contains similar elements to those found in a SOA, i.e. a provider, a consumer and a registry.

In chapter 4 we described how we decided to provide our MPS through an enterprise information portal. In line with this decision, we argued that there was a need to standardize our design on the Portlet Specification (Abdelnur and Hepper 2003). We argue that hereby we made an implicit choice for a service oriented architecture. An enterprise information portal acts as both a service provider and a service registry. The service consumer is the browser that is used to access the portal, and services are provided in the form of portlets. Following the Portlet Specification, each portlet application should be accompanied by a file, named portlet.xml, that represents a service description.

To come back to our question on the difference between building blocks and services in our MPS, we refer to the concept of a container (Farley and Crawford 2005, Genender et al. 2007) (see chapter 5). We adhere to Sun Microsystems' definition, it defines a container as an entity that provides life-cycle management, security, deployment, and runtime functionalities to software components[25]. Papazoglou and Georgakopoulos (2003) extend the SOA, introduced by Endrei et al. (2004), with a "service composition layer" that provides functionalities such as the control of service execution, enforcement of business rules and data flow management. Based on this extension, we conclude that a container acts as an essential element for the rapid, low-cost composition of a distributed information system such as our MPS. Hence, to answer our question, we argue that the services of our suite should implement a container interface while building blocks do not. The typical front-end containers provided by our programming platform are the portlet, applet and Java Web Start containers. We argue that the services of our MPS should be designed as portlets, applets, or Java Web Start applications that are provided by, and loosely coupled through an enterprise information portal.

6.2 Map service

6.2.1 Function of Map service

The *Map* service is intended to support the need for (dynamic) geovisualizations in mainport planning (requirements 2 and 3). Geographic maps are used commonly in mainport planning as we observed in our exploratory case studies (chapter 2). Geographic maps are used to

[25] See: http://java.sun.com/javaee/reference/glossary/index.jsp#88608.

analyze a mainport area, for example to gain insight into the available infrastructure and facilities, or to gain insight into the relation between noise levels and activities in the mainport. Geographic maps are also used as a shared viewpoint and writing pad in joint design sessions, where mainport planners sketch their creative ideas on paper maps. The Port of Rotterdam has developed an advanced map viewer called RIV, which functions as an electronic atlas for many types of geographic information available in the port. Despite their reliance on maps, we also observed that mainport planners are reluctant to use advanced GIS systems such as RIV. Instead of making use of RIV, mainport planners in the PoR are used to ordering printouts of specific maps from the design and drawing department. We believe that there is a discrepancy between the needs of mainport planners and the technology they have currently at their disposal. Mainport planners are usually not GIS experts. At the same time we have seen that during the past few years Google has managed successfully to make GIS available to the general public. GIS is no longer only for experts, and millions of people can now find hotels, restaurants and other location based facilities while interacting with a web-based map. Today the users of geographic maps also become the authors and distributors of their own maps. They can sketch on and annotate existing maps, they can overlay their own information on top of existing maps, and they can share their creations over the Web.

The *Map* service provided by our MPS was designed specifically to address the needs of mainport planners. In close cooperation with experts in the PoR we iteratively rethought and redesigned the concept of an interactive map viewer. The *Map* service is a web-based interactive viewer for both static and animated geographic maps as is shown in Figure 51 and Figure 52. Below we describe the design considerations of the *Map* service in detail.

- Importantly the *Map* service should support the design and evaluation of alternative solutions. The *Map* service is not meant to be an all in one viewer for gaining insight into the current situation, it is meant to be a lightweight creativity tool which supports rehearsing the future of a mainport. Mainport planners should be able to have their information, model outcomes and design sketches quickly translated into layers of GIS maps, which can be used by individual experts and multi-disciplinary planning teams. Central to the creation of maps is the map-file which specifies which shapes, bitmaps and location based data should be used in the map.
- Related to the previous point, the *Map* service should support distributed people with using distributed information resources. It should make geographic information available to mainport planners who work dispersed over the mainport's organization and who require web-based access to, visualization of and interaction with completely different types of information. The information displayed can be of many types, related to both the content and the process of mainport planning. Information related to the content could be land use information, safety contours, noise budgets, traffic densities i.e. anything that has a location. Information related to the process could be project boundaries, locations of people's offices, etc.. In other words, we want a map to be like a web page: viewable by many dispersed users, and clickable to "hyperlink" to remote information soures.
- The *Map* service should support the distinction between providers of maps and users of these maps. The providers of maps are GIS analysts who have expert knowledge in creating quality cartographic material for which they use advanced GIS technology. The users of maps are mainport planners who commonly lack the skills and expertise of GIS

analysts; therefore they need to be provided with easy to use, interactive map visualizations.

- The *Map* service is designed with the animation of temporal information in mind. The *Map* service makes use of the Map library which we introduced in chapter 5 and thereby it provides functionality for binding layers of geographic information to time intervals. Hence *Map* service can be used to display how a mainport changes over time geographically. Furthermore, because the *Map* service is also based on the visualization library introduced in chapter 5, it can be used for animating simulation outcomes directly on a map. Different time advance mechanisms should be supported by the *Map* service. In Figure 51 two possibilities are shown. A time interval slider is a GUI component which a user can use manually to specify a certain time interval using sliders. Alternaitvely animation controls can be used for starting, pausing, stopping etc..

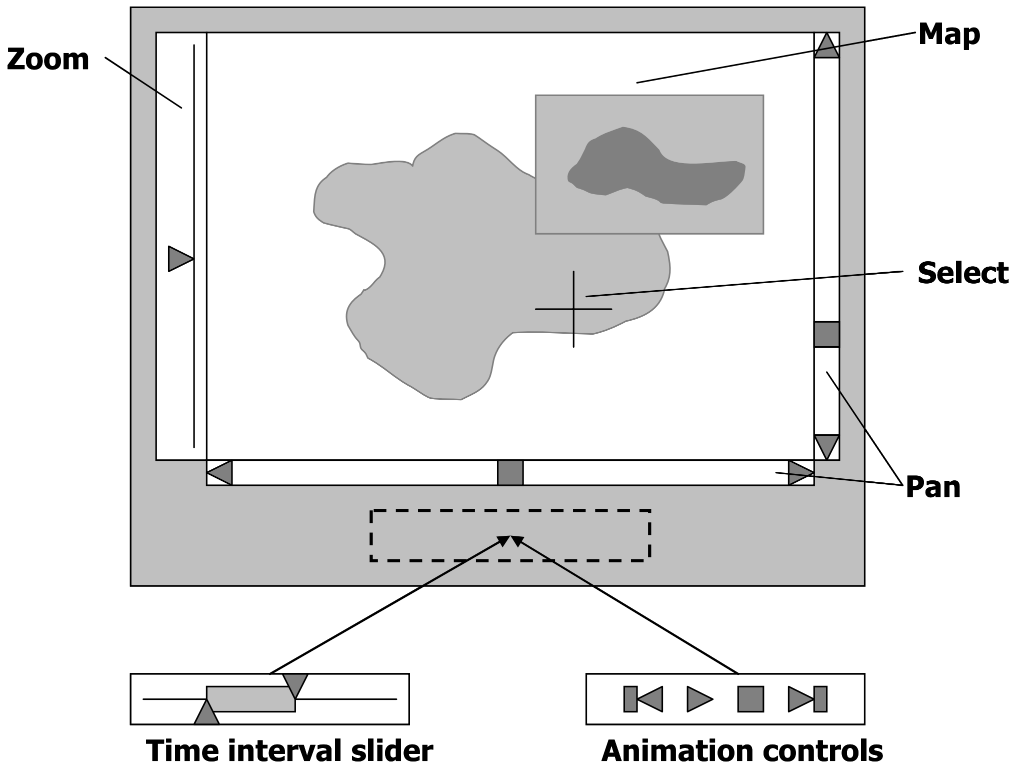


Figure 51: Map conceptual design.

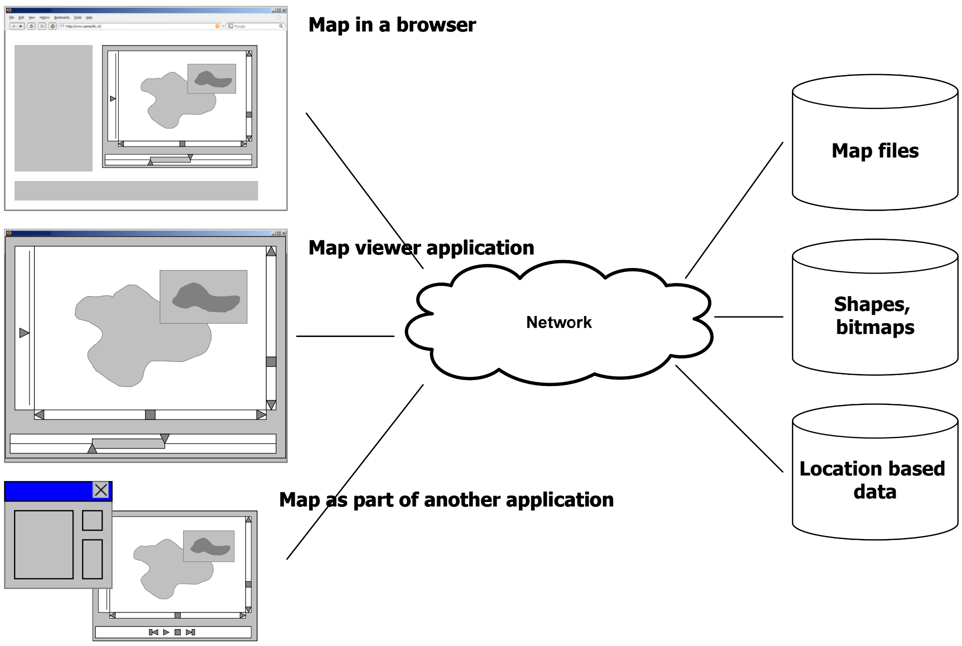


Figure 52: Using Map in different ways to visualize and interact with distributed information.

6.2.2 Using Map service in mainport planning

An Idef0 diagram (Menzel and Mayer 1998) of how *Map* is intended to be used in mainport planning is shown in Figure 53. During a planning process a planning team requires maps that are custom made to address specific issues in multi-disciplinary team meetings. Mainport planners know what data they want to display on the map and they might have some sketches that they want to be processed. They contact a GIS analyst to prepare a map based on their provided inputs and requirements (A1). The GIS analyst uses his or her own GIS tools for this task. He or she creates a map file (see chapter 5) and the required bitmaps and shape files. Possibly he or she also specifies links to data that is available in the organization's databases, and he or she might incorporate dynamic models that should be animated on the map.

Next the specified map is used in the *Map* service by the planning team (A2). They use the *Map* service to analyze the mainport area under investigation, to discuss bottlenecks and to search for alternative solutions. To document their findings they write down their conclusions and comments. Furthermore the mainport planning team may draw up design sketches of alternative solutions. This can be done digitally in *Map*, or on a paper map. Finally they might take these sketches and ideas back to the GIS analyst, such that this information can again be processed into an "official" GIS map which is stored in the corporate databases.

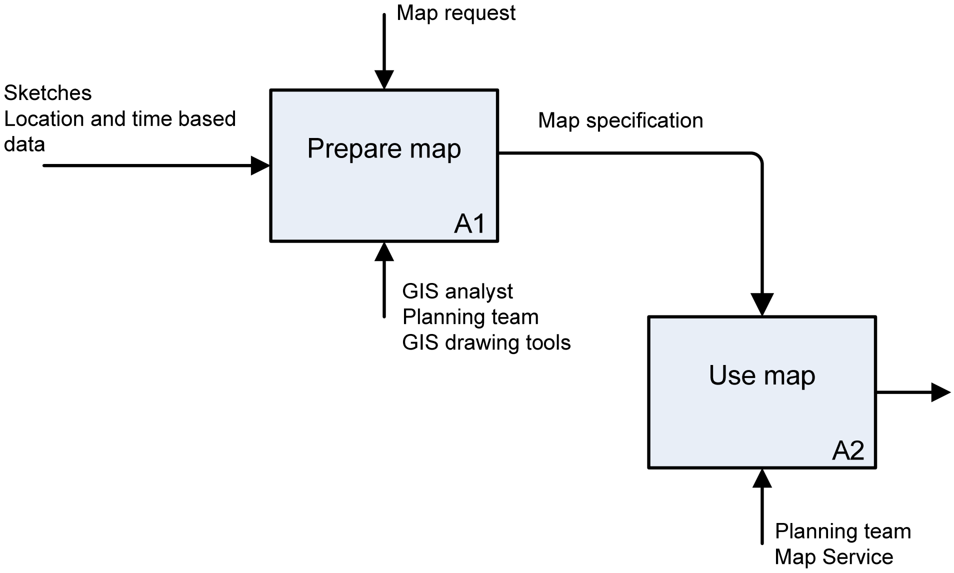


Figure 53: Idef0 diagram of Map usage in mainport planning.

6.2.3 Specification of Map service

The user interface components of *Map* are shown in Figure 54. By example, the figure shows a map of an area in the Port of Rotterdam. The map panel is an interactive visualization of a geographic map. It displays the layers specified in the map file. The map can be zoomed in and out, and the map can be panned in all directions. The second component is the time slider, which represents an interactive timeline. The user can specify a time interval using the time slider by sliding the buttons over the timeline. When a new time interval is specified, the time slider fires an event (TIME_INTERVAL_EVENT) to the map panel. Next the map panel updates the dynamic layers and shapes according to the specified time interval. For example, as is shown in the figure, the time slider is used to select ongoing and planned project boundaries in the PoR. Each project is displayed as a yellow shape and is labeled with relevant project information, such as the name of the project leader.
The time slider and the map panel are loosely coupled, because the only dependence between both components is the event messages. Therefore it is possible to replace the time slider by another time advance mechanism such as a simulation clock (simulator). This enables the use of a Map panel for animating simulation outcomes on the map.

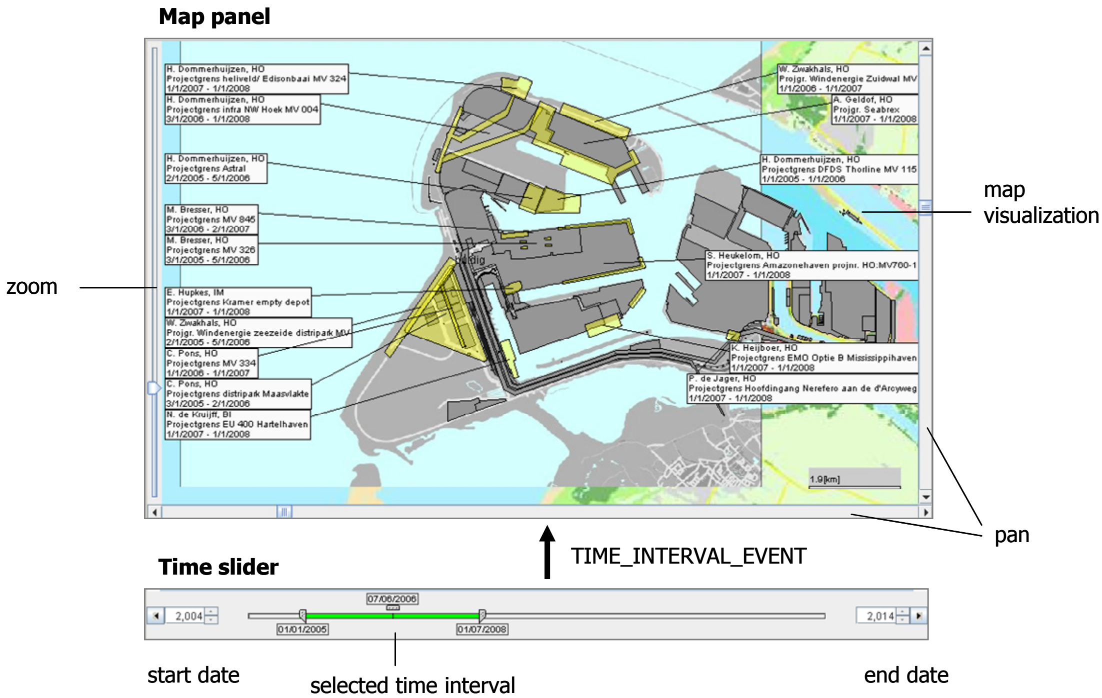


Figure 54: User interface components of Map.

Map builds on top of the geovisualization library (see chapter 5). The architecture of *Map* is shown in Figure 55. *Map* uses the geovisualization library to render maps. The layers that compose the map are specified in a map-file. This file points to other resources required to render the map, e.g. raster and vector images. *Map* can be provided to the user in different ways, using a range of thin and rich clients.

- Map Java Web Start / desktop client: this is a panel which is meant to run as (part of) a desktop application. There is no client-server communication as all the code runs on a single machine. However, the desktop client can access remote resources such as raster and vector images on a file server.
- Map Applet: this is a Java applet which provides the same functionality as the desktop client, but with the ability to display maps which are rendered on a remote server. Consequently maps that are rendered on the server can be combined with data rendered on the client. Furthermore the applet can be embedded in a webpage and as such it can become an integral part of a portlet application.
- Map AJAX client: this is a fully AJAX enabled client that runs in a web browser similar to e.g. Google Maps (http://maps.google.com). In contrast to the applet it can not do any client side rendering except for placing markers and pre-rendered images on the map. Maps are rendered on the server and the result is displayed in the client.

Server-side rendering is performed by a MapServer servlet. The servlet runs on a web server and can be queried to render a specific map. The rendered map is a bitmap image which is streamed back to the client.

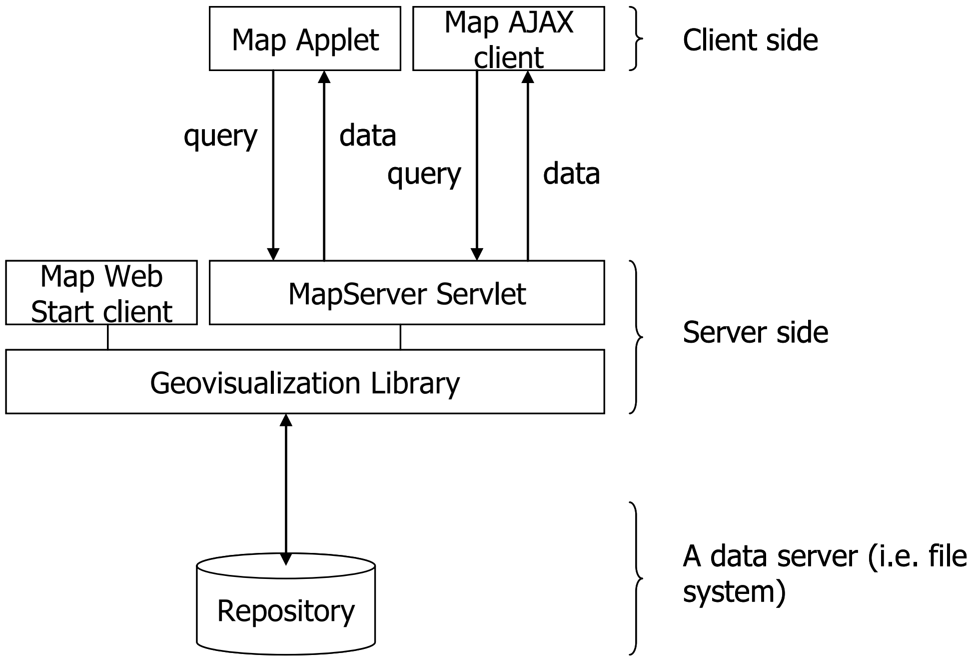


Figure 55: The architecture of Map.

6.3 Sketchbook service

6.3.1 Function of Sketchbook service

The *Sketchbook* service is meant to support indexing, filtering, viewing and interacting with information used and produced during a mainport planning process (requirement 4). In mainport planning an abundance of information is used and produced. A substantial amount of information exists in the form of images such as geographic maps, diagrams, and graphs. Other forms of multi-media may also be used such as photographs, movies and animations. Typically the available information is stored in distributed databases which fall under the responsibility of different departments and individuals. As we observed in our exploratory case studies (chapter 2), it is not easy to manage and work with the abundance of information used and produced in mainport planning.

Sketchbook uses the metaphor of a paper sketchbook or scrapbook. A conceptual design of *Sketchbook* is shown in Figure 56. Below we describe our design considerations.

- Sketching, drawing up ideas and annotating images are common tasks for designers and mainport planners as we found in our exploratory case studies. In a paper sketchbook all pages are connected, i.e. one needs to flip the pages to go from one sketch to another. Yet, this is not how designers like to browse through their information (Keller et al. 2004, Keller et al. 2006, Steenbergen et al. 2002, Van der Lugt 2001). Designers like to create many different perspectives on the same information and they like to combine and compare very different types of information. For example, they may want to put a design sketch next to a photo of the current situation and simultaneously see a 3D model on a computer screen. Simply by visiting a design studio one can observe how designers really work with images, multimedia and other types of information. They put images on the table, stick them to the wall, show them on an overhead projector, use a computer to generate 3D images, or do all this simultaneously. They create storylines by putting images behind each other and annotating them. They make choices by comparing images spread out on the table and marking them with sticky notes. They create structure by creating stacks of images in different categories. Designers rip the pages from their sketchbook, but this makes it difficult to keep track of and organize the information in long term mainport planning processes.

- *Sketchbook* is meant to offer mainport planners the freedom they need to combine and compare information of many types creatively, without the danger of loosing track of information during the mainport planning process. *Sketchbook* does not have pages in a fixed order, but provides many different viewpoints on information. Information can be displayed as a photo album, on a geographical map used as a backdrop, on a time line, or in another useful way. Furthermore information can be annotated using a blog or wiki to make clear which of the thousand words that an image is supposed to substitute for were meant by its creator.
- In a sense *Sketchbook* resembles a file manager such as those found in the major operating systems. The information which is made available by *Sketchbook*, such as design sketches, geographic maps, and photos, are files which reside inside file systems. However, as we argued in chapter 4, the hierarchical structure of file systems is not optimal for working creatively with information. Sketchbook links to files, which are stored in the dispersed file systems of a mainport's IT infrastructure, and uses virtual folders (Jones 1999, Cole et al. 2000) to present and organize this information in mainport planning projects.

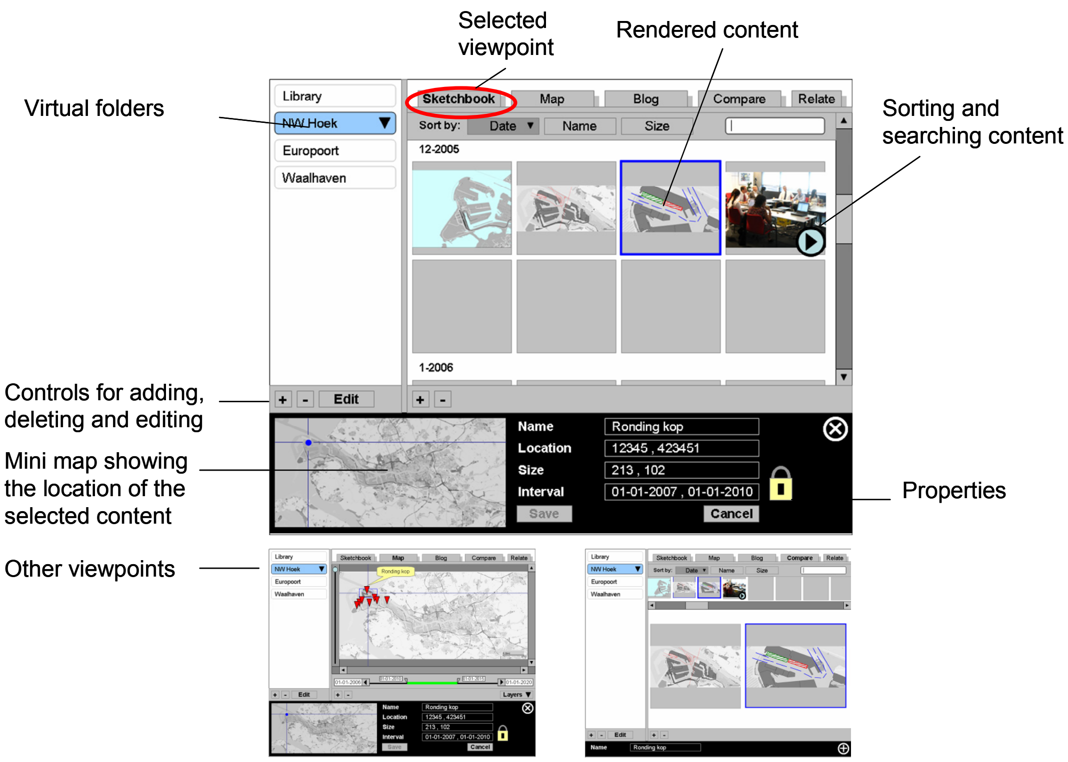


Figure 56: Sketchbook conceptual design.

6.3.2 Using Sketchbook service in mainport planning

An Idef0 diagram of the process of using *Sketchbook* in mainport planning is shown in Figure 57. We shall refer to content as the information stored in sketchbook: images, maps, sketches, photos, and other types of information that is available in a digital format. In fact, the content stored in Sketchbook usually consists of computer files. The *Indexer*, *Virtual folders*, *Viewpoint*, *Launcher* and *Blog* are subsystems of *Sketchbook* that are explained in the next section.

The first step in using Sketchbook is adding content to it (A1). Initially a planning team can add relevant content to make a start with the planning project. During the planning project new content is added to sketchbook. Next the content needs to be organized and structured which is done by the individual team members (A2). Next the content can be browsed from many different points of view (A3). Finally the content can be opened such that it can actually be used, or the content can be annotated.

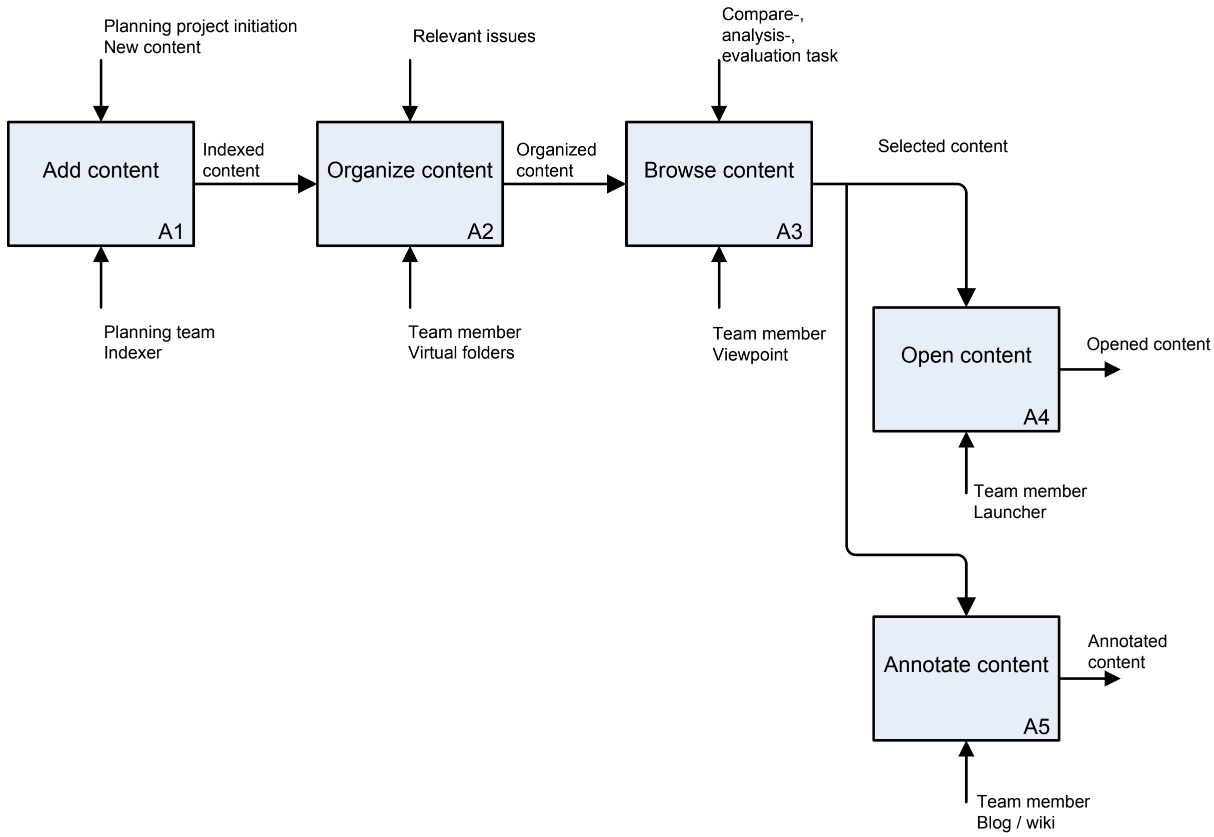


Figure 57: Idef0 diagram of Sketchbook usage in mainport planning.

6.3.3 Specification of Sketchbook service

Figure 58 shows the user interface components of *Sketchbook*, which is made available as an AJAX enabled portlet in a web portal. The *Virtual folders* component displays the organization of content, i.e. the files that are made available by *Sketchbook*. We argue that *Sketchbook* should support two types of virtual folders (Figure 59).

- Virtual folders can be used to automatically filter content based on meta-data. For example, mainport planners could specify a virtual folder that shows all geographic maps which relate to a specific period in time. In that case a virtual folder does not directly link to information, but it stores a query which is used to execute “a search on demand”, i.e. each time the folder is opened.

- Virtual folders can be used to group content on virtual stacks. Keller et al. (2004) state that designers often use stacks of images to group related information. The problem with physical images is that an image can be only on one stack at a time, but using virtual folders in *Sketchbook* it is possible to put the same image on multiple stacks at simultaniously, i.e. several virtual folders can link to the same content. In this case a virtual folder keeps an index of files to which it links. This index is created manually by the mainport planners depending on how they wish to group content.

Viewpoint selection is a list of available viewpoints. A selected viewpoint appears to the right. For example, when selecting 'Album' the content appears as a photo album, or when selecting Map the content appears on an interactive geographic map, which is actually an instance of the *Map* service. From a *Viewpoint* a user can select content, just as he or she would in a file manager (such as Windows Explorer®). The Selected content information component displays the name of the selected content and can be expanded to display meta-data such as the content's creator, modification date, URI, and other relevant information. By double clicking a user can open the selected content. As we stated before *Sketchbook* has similarities with a file manger, however in contrast to a common file manager, *Sketchbook*:

- does not use a directory tree to organize its content
- provides viewpoints which are useful in the mainport planning domain
- aggregates information from different locations in a single visualization
- is made available in a web browser and thus is a service which is shared among multiple users

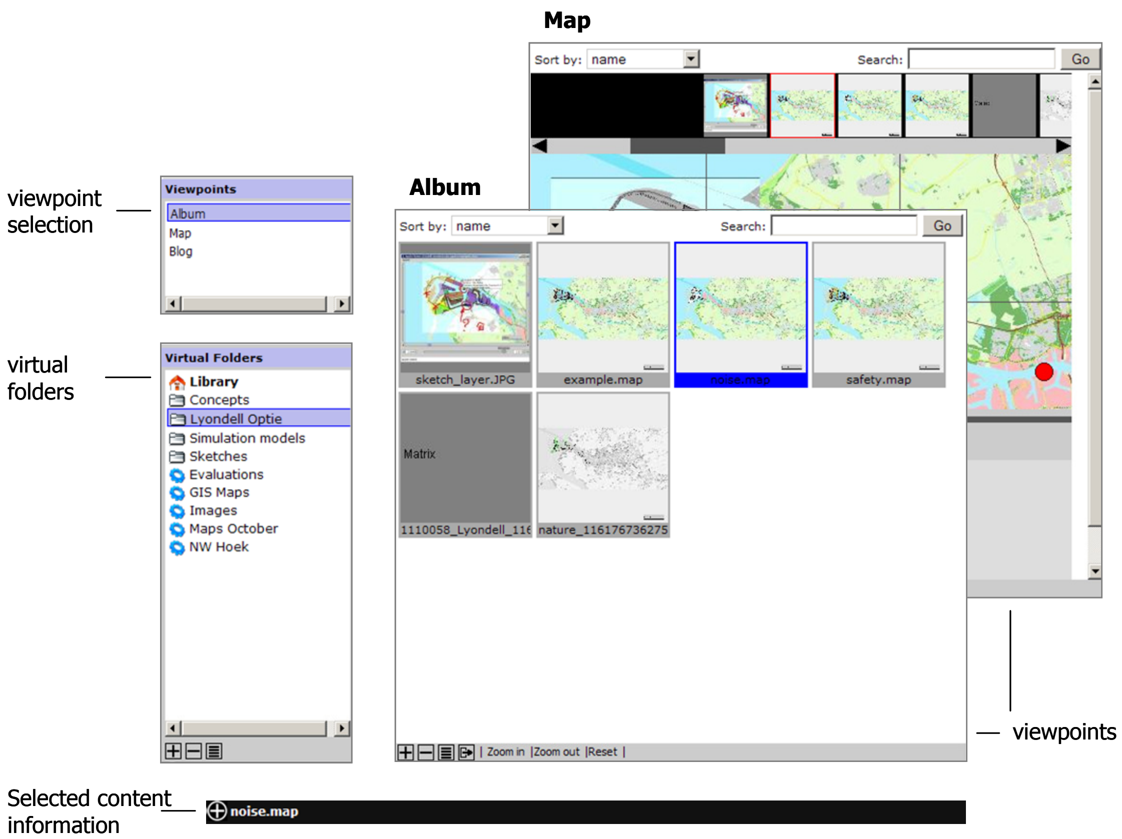


Figure 58: User interface components of Sketchbook.

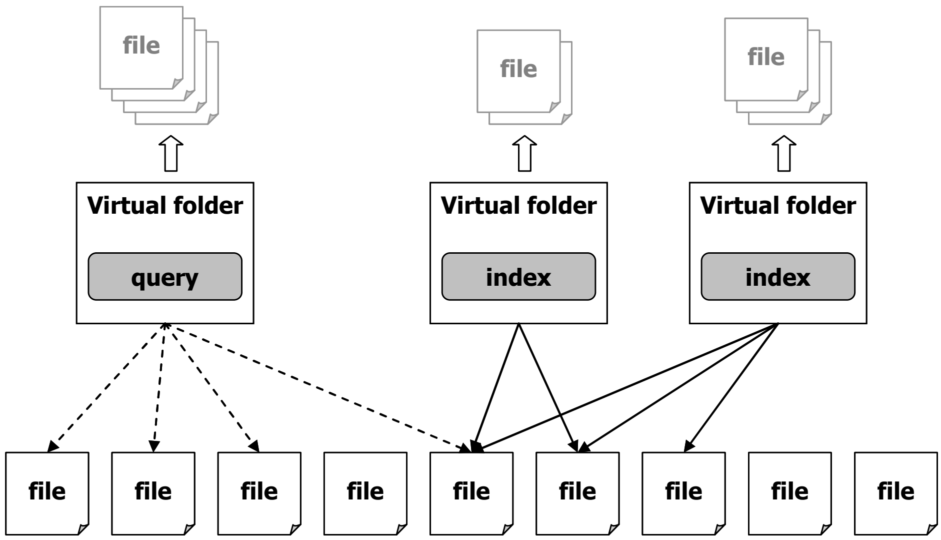


Figure 59: In Sketchbook virtual folders are used to organize information, i.e. files. The virtual folders either use a query or an index to link to files.

Figure 60 displays the architecture of *Sketchbook*, which follows the steps for information visualization: access raw data, filter the data, map visual structures onto the data and create views (Card et al. 1999) (see chapters 3 and 4). Several subsystems can be distinguished.

- Indexing Service: creates an index of content available in distributed data sources such as Samba servers and FTP servers; the indexing service has its own user interface for managing remote data sources.
- Virtual folders: organize the indexed content in virtual folders; two types of virtual folders can be distinguished namely folders which store a query on the content, e.g. all content from November 2006, and folders which link to arbitrary content as set by a user i.e. to create virtual stacks of information.
- Renderer: renders a thumbnail image of the indexed content.
- Viewpoint viewer: combines the rendered thumbnail images with other visual structures to create a specific view on the content; different viewpoints such as AlbumViewpoint and MapViewpoint can be implemented.
- Launcher: opens the content in a specified application depending on the type of content, for example a geographic map is opened in *Map*.

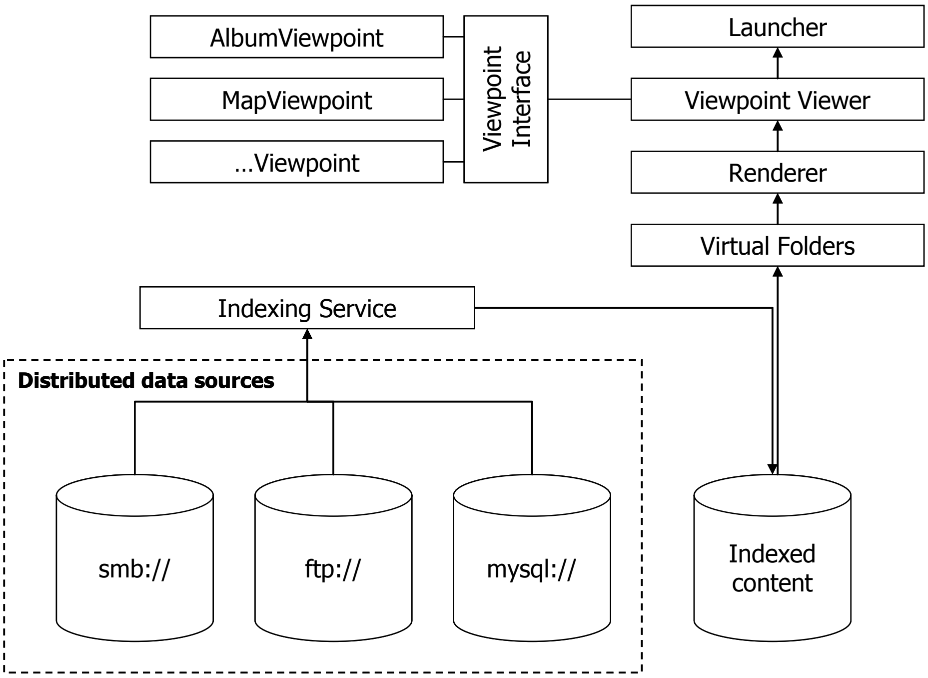


Figure 60: Sketchbook architecture.

6.4 Matchbox

6.4.1 Function of Matchbox service

Matchbox is a service meant to support the evaluation of alternative solutions in mainport planning. The original concept of *Matchbox* was the evaluation matrix which the area planners in the PoR used as a checklist in the evaluation of alternative industry types (see chapter 2). *Matchbox*, which we designed in close cooperation with the mainport planners in the PoR, goes beyond being a checklist; it visualizes the status of a multi-disciplinary evaluation process in which numerous aspects need to be compared.

The conceptual design of Matchbox is shown in Figure 61. Three main elements can be distinguished: a context, i.e. the subject of the evaluation, aspects to consider in the evaluation, and alternatives that should be evaluated. Each alternative-aspect combination needs to be evaluated to determine whether the combination is possible or desirable. Based on the individual alternative-aspect evaluations it can be determined if an alternative should be accepted or rejected. For example, in a port the context could be a parcel of land for which a suitable use should be found; the alternatives would then be possible industry types for that parcel of land; and the aspects could be e.g. square meters of space, traffic load, and noise levels.

Below we describe our design considerations.

- *Matchbox* should support different domain experts in the evaluation of a large variety of aspects and alternatives. The evaluation of alternative solutions in mainport planning is a complex and information intensive task. For example, land use planning in the PoR requires mainport planners to consider the available space, the accessibility, safety, pollution level and commercial attractiveness of a port area. Each of these aspects can be broken down into more detailed sub-aspects. For example, accessibility can be broken down into accessibility by road, rail, waterways and pipelines etc.. Consequently a large amount of expert knowledge and information needs to be combined and compared to make an evaluation of alternative solutions.

- *Matchbox* should support both quantitative and qualitative considerations in the evaluation of alternative-aspect combinations. The knowledge and information to make an evaluation is related to both possibilities and desirabilities. In some cases there are hard quantitative constraints which may not be violated by an alternative solution. For example, a hard constraint may be stated as the available square meters for a customer, or safety regulations. Sometimes counter measures can be taken to overcome hard constraints. One could, for example, construct a noise wall to stay within the legal limits for noise production. Yet in other cases the constraints are more related to what is desirable instead of what is possible (or impossible), i.e. the constraints are more qualitative in nature. For example, it could be commercially desirable to cluster port companies that can exchange half fabricates, to facilitate this exchange.
- *Matchbox* should support a multi-disciplinary team of mainport planners in making a distinction between accepted and rejected alternatives. We do not follow Malczewski (1999) who advocates the use of multi-criteria analysis for spatial planning. Although it is tempting to formalize the evaluation of alternative solutions based on a scorecart with weight factors (Cross 1994), we argue that a different approach is needed. The use of weight factors suggests that qualitative aspects can be quantitatively evaluated. We doubt this is so: How can one quantitatively express how important safety is compared to commercial attractiveness? During our case study at the PoR (chapter 2) we learned that the evaluation of alternative solutions is, for a substantial part, a negotiation process between the involved mainport planners. Usually there is not a single best solution but a whole range of satisficing solutions. Mainport planners indicated that they suffered from cognitive constraints during the evaluation of alternative solutions because of the abundance of information involved. *Matchbox* is used to visualize how a range of alternative solutions compares, based on the qualitative and quantitative information provided by domain experts. As an evaluation of alternative solutions progresses, domain experts bring in an increasing amount of information regarding relevant aspects. *Matchbox* visualizes how far an evaluation has progressed and where there are still open issues that need to be addressed. A similar concept is described by Walker (2000), who introduces a scorecard that does not use weight factors.

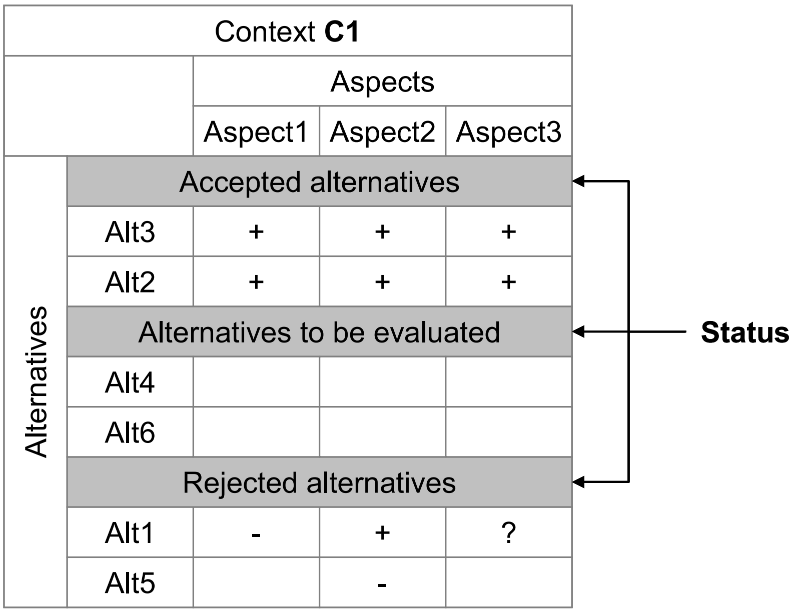


Figure 61: Matchbox conceptual design.

6.4.2 Using Matchbox service in mainport planning

An Idef0 diagram of using *Matchbox* in mainport planning is shown in Figure 62. The first step (A1) when an evaluation activity starts is to specify Matchbox's content:

- the context of the evaluation: what it is about
- the alternative solutions that should be evaluated
- the aspects that are considered relevant in the evaluation
- the key values of alternative-aspect combinations, e.g. produced decibels for a noise aspect

Next the individual team members evaluate the alternatives per aspect (A2). This means that each individual team member evaluates the alternative solutions from their own field of expertise. For each combination of an alternative and aspect he or she must indicate whether the combination is possible or desired, or impossible or undesired, or whether further research is required. The team member must also motivate his or her choice in text.

Finally the complete team comes together to have a look at the complete matrix of alternatives and aspects. Now, they have to decide, as a multi-disciplinary team, which alternatives they will approve and which they will reject.

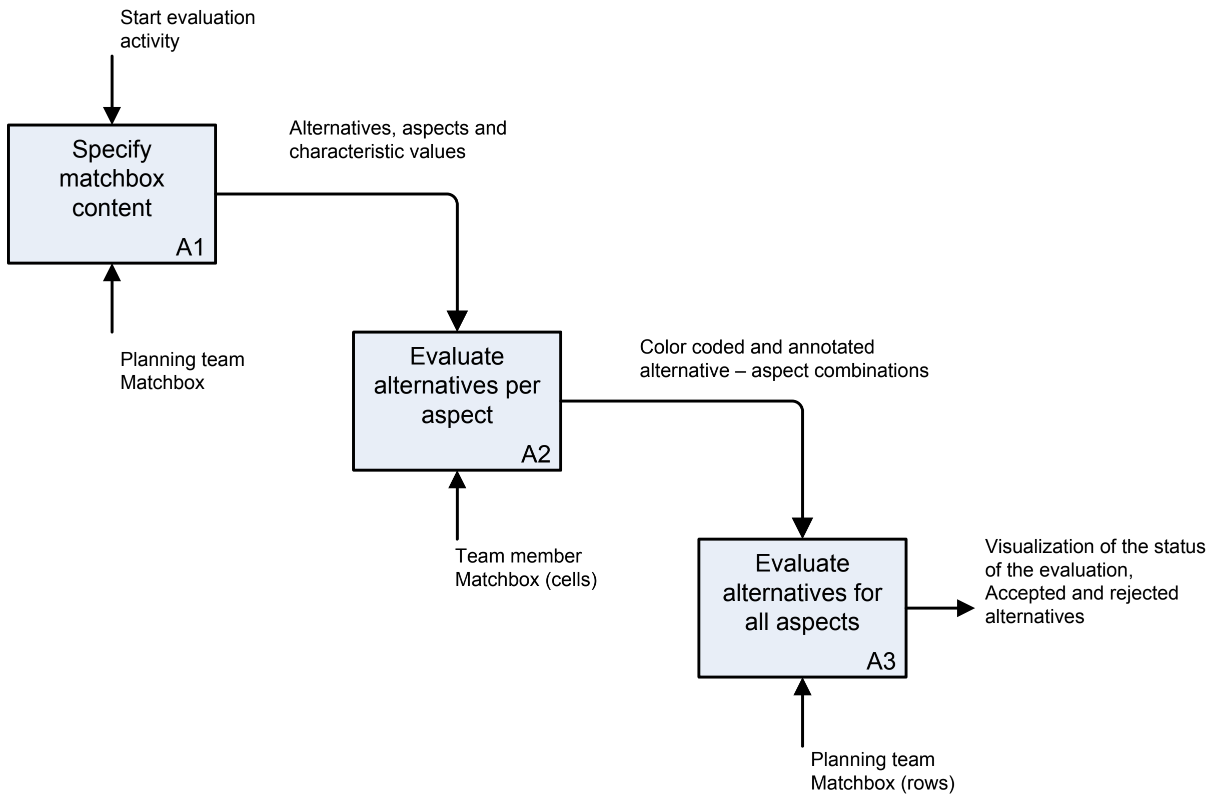


Figure 62: Idef0 diagram of Matchbox usage in mainport planning.

6.4.3 Specification of Matchbox service

The user interface components of *Matchbox* are shown in Figure 63, using an example releated to an evaluation of land use in the PoR. *Matchbox* is presented as an interactive table where the rows contain the alternative solutions and the columns contain the relevant aspects for the evaluation. Each cell, i.e. combination of alternative and aspect, contains a key value or an expression. For example, when evaluating possible customers for a lot in a PoR, a key value for a deep-sea company's required space can be expressed as a range of 60-120 Ha.

During the evaluation of alternatives per aspect, i.e. process A2 in Figure 62, domain experts use color coding to express their preference regarding individual alternative-aspect combinations in their field of expertise:

- green: possible or desirable
- yellow: unclear / needs further research
- red: impossible / undesirable
- white: not evaluated

An expert can also type "behind" each cell comments as to why he or she applied a certain color coding.

The evaluation of all aspects by a multi-disciplinary team of domain experts, i.e. process A3 in Figure 62, is supported by three distinct categories:

- approved: alternative solutions that fit the context
- under evaluation: alternative solutions that should still be evaluated

- rejected: alternative solutions that are rejected for the specified context

These categories are used to group the available alternatives. In the beginning of the evaluation all alternative solutions are in the category “under evaluation”. As the evaluation progresses and more information becomes available, the alternatives are either moved to the “approved” category or moved to the “rejected” category. This is done manually by a team member based on the outcome of a mainport planning team negotiation. Finally no alternative solutions should remain under evaluation.

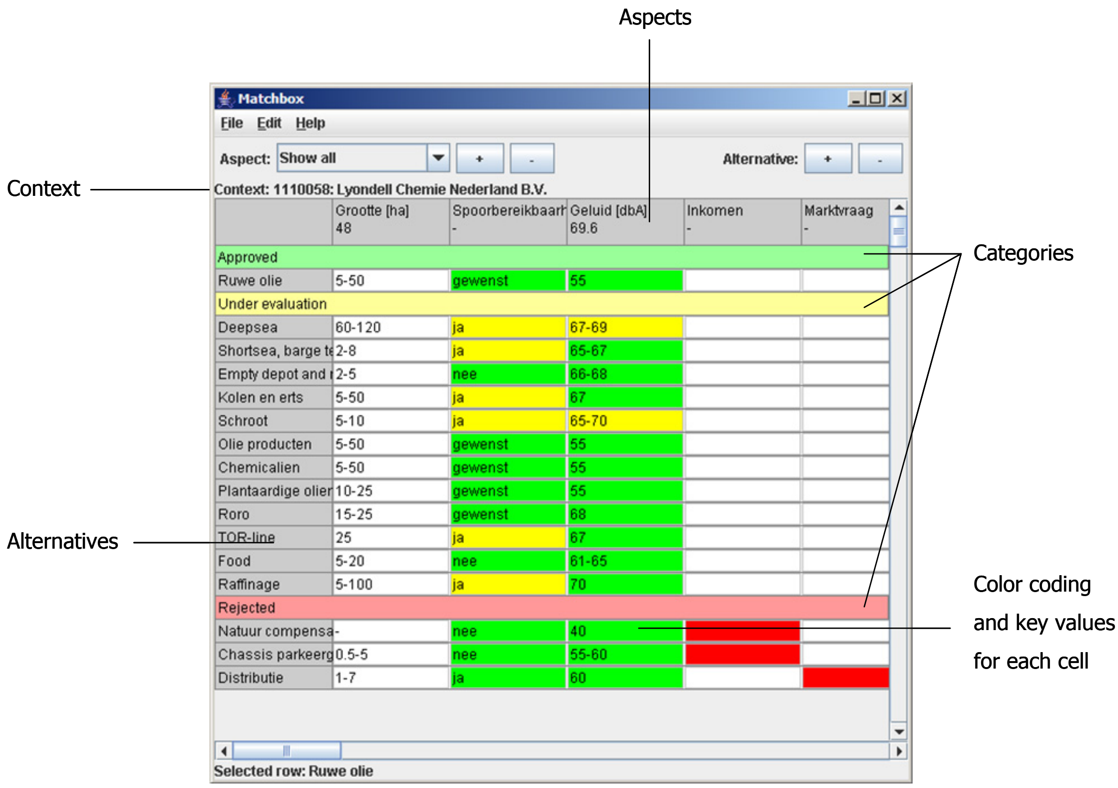


Figure 63: User interface components of Matchbox.

6.5 *Aspect Explorer*

6.5.1 Function of Aspect Explorer service

Aspect Explorer is a service that is meant to support making a choice between viable alternative solutions. Whereas the outcome of *Matchbox* is a list of viable alternative solutions, a planning team still has to choose a preferred solution. When there is only a small number of viable solutions and relevant alternatives, the team can probably easily decide which solution they prefer. As the amount of alternatives and aspects becomes larger it becomes more difficult to make a choice among alternative solutions.

The conceptual design of *Aspect Explorer* is shown in Figure 64. In essence Aspect Explorer is an exact implementation of Spence’s neighborhood explorer (2001). Spence demonstrated how this visualization concept can be used to make a choice among houses that are for sale without the need to quantify the available information. In *Aspect Explorer* radial axis are used to represent the aspects that are considered in making a choice for a preferred solution. The alternative solutions are presented on each axis. The alternative solutions are sorted top to bottom for each aspect with the most preferred alternative on top and the least preferred alternative at the bottom. When sliding one of the alternatives to the centre of the diagram, one can see how it relates to the other alternatives for each aspect.

We chose to adopt Spence's visualization concept to support making choices in mainport planning, because it provides a shared visual representation that allows for a purely qualitative comparison of alternative solutions. *Aspect Explorer* can be used to show the preferences of domain experts regarding viable alternative solutions in a single (interactive) image. The result is that it is possible to show specifically how the mainport planning team as a whole sees the solution space.

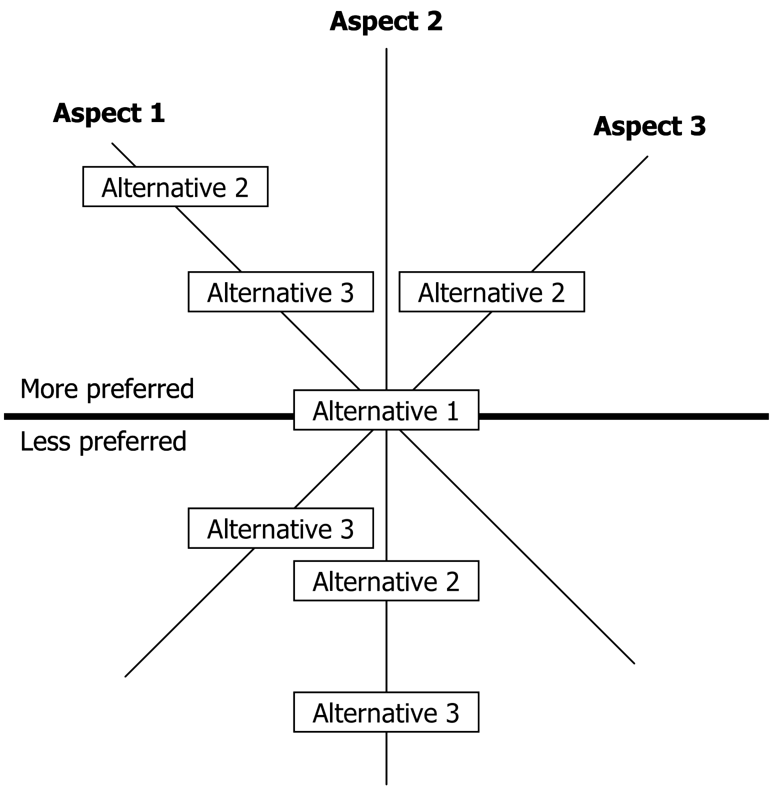


Figure 64: Aspect Explorer conceptual design.

6.5.2 Using Aspect Explorer service in mainport planning

An Idef0 diagram of using *Aspect Explorer* in mainport planning is shown in Figure 65. First the planning team specifies the content of *Aspect Explorer*: which alternative solutions should be considered (A1). Most likely the alternative solutions and aspects are directly taken from *Matchbox*. The selected aspects are probably also taken from *Matchbox*, although probably only those that are considered as most crucial for making a choice. Next, each team member makes a ranking of the selected alternatives for the aspects in which he or she is the expert (A2). Making the ranking simply means putting the alternatives in order of preference, the most preferred alternative on top; and finally the planning team uses the *Aspect Explorer* in a meeting in which they discuss the different alternatives and try to choose the most promising alternative (A3).

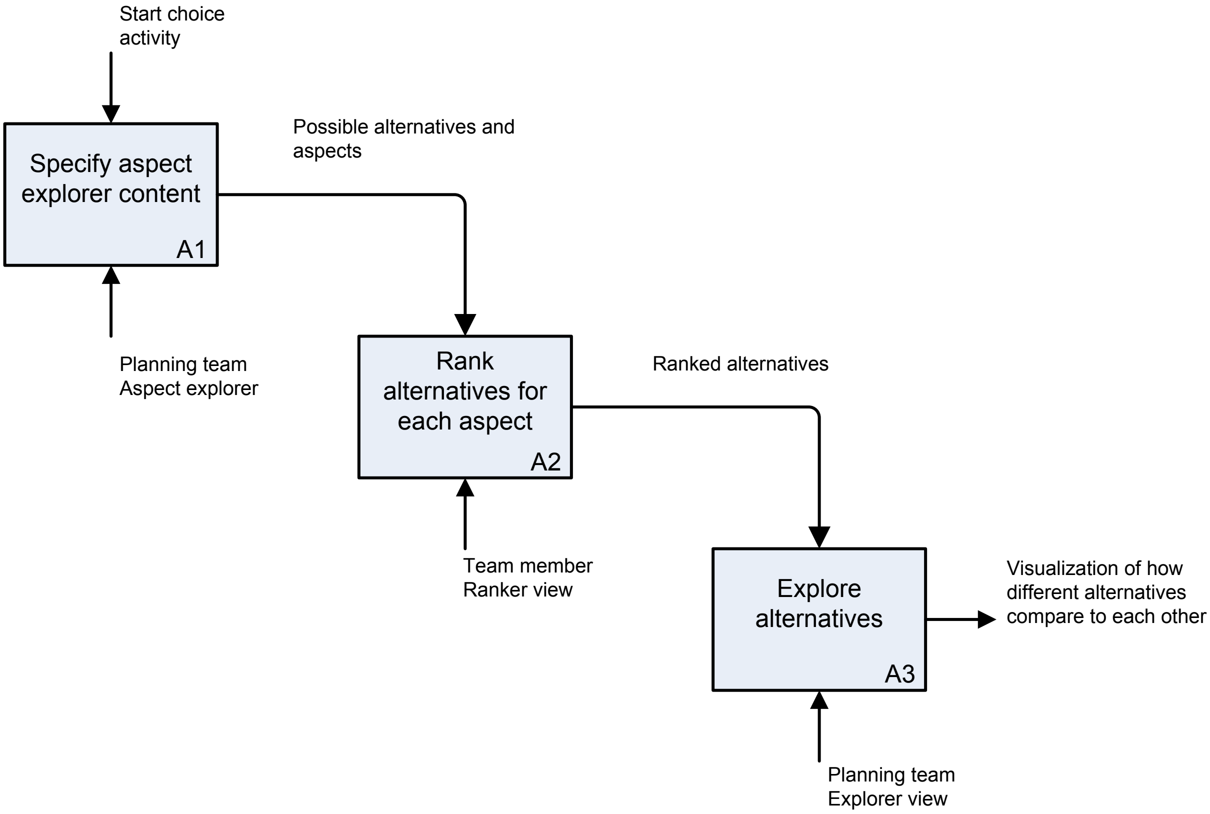


Figure 65: Idef0 diagram of Aspect Explorer usage in mainport planning.

6.5.3 Specification of Aspect Explorer service

Figure 66 shows the user interface components of *Aspect Explorer*. There are two main views: ranker and explorer. In ranker an aspect is selected. Next the specified alternatives can be ranked for the selected alternative. The alternatives are placed (dragged) on the vertical axis where the most preferred alternative is the topmost and the least preferred alternative is the lowest. Only the position relative to the other alternatives is considered, not the absolute placement on the panel or the distance between the alternatives. Each individual expert should make a ranking for the aspect in his or her field of expertise.

Next, one can switch to the explorer view. In this view the aspects are shown as radial axis and the alternatives are placed onto these axis according to their ranking. When the user clicks on an alternative, then it slides to the center. This way a team of planners can sequentially put alternative solutions in the center and see how they compare to the other alternatives. Alternatives above the red line “score” better on the respective aspects than the centered alternative and vice versa. Consequently, the explorer view provides a quick overview of the (qualitative) preferences of the individual experts.

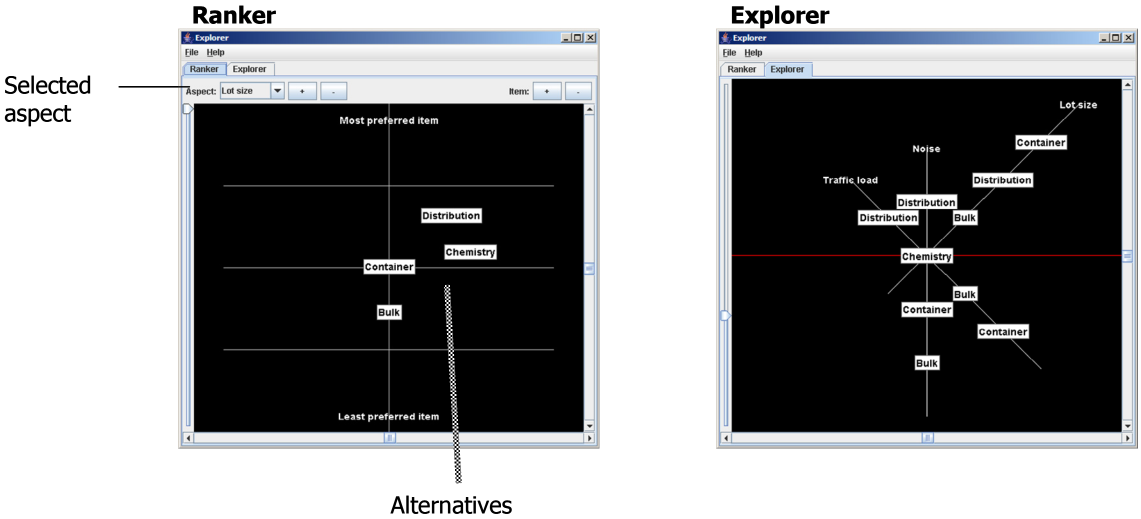


Figure 66: User interface components of Aspect Explorer.

6.6 Process support services

In addition to supporting the content of mainport planning, the activities such as analysis, design, evaluation and choice (requirement 1), we introduced services to support the process of mainport planning (requirement 5). As we described in chapter 2, a mainport planning project takes months of work and involves a multi-disciplinary team of experts. Each expert fulfills a different role in the planning project, and during a planning project new people may be introduced, people will leave the project, or people can assume different roles. Furthermore, a planning project is structured along meetings and during each meeting agendas are set, actions are agreed upon and decisions are made. We learned from our case in the PoR (chapter 2) that the actors involved in mainport planning find it difficult to keep track of the planning process. Hence we identified a need for services to support making annotations, comments, and for keeping track of actions, decisions and meeting minutes. Furthermore we identified a need to support actors in different roles, with different information needs.

The process support services are designed to address these needs. Figure 67 shows a conceptual UML class diagram of the information provided by the process support services. As mainport planning is organized in projects, we choose the project as the center of our design. Each project has a name, a project description, a project manager, a start date, an end date and a time interval for which the plan is made. Furthermore each project involves a number of participants: the team members. A project schema represents the different phases which eventually lead to the completion of the mainport plan. Examples of project phases are: startup, analysis, design, reporting. The phases that are specified for each project can differ depending on the complexity of the project and the amount of experts involved.

We distinguish two ways of looking at a project: issues and meetings. An issue refers to specific topics which are addressed during a project. For example, a planning team in the PoR may create an issue concerning a specific terrain in the port. All information regarding this terrain will then belong to this issue. In fact, each issue can be seen as a link to the content of mainport planning: the information related to analysis, design, evaluation and choice.

Meetings represent the development of a project through time as projects move from meeting to meeting. A number of agenda items are addressed during each meeting. Furthermore the participants of a meeting define actions that should be completed at a later stage within the project, and they make decisions that influence the course of the planning project. Finally each meeting should be documented in the meeting minutes.

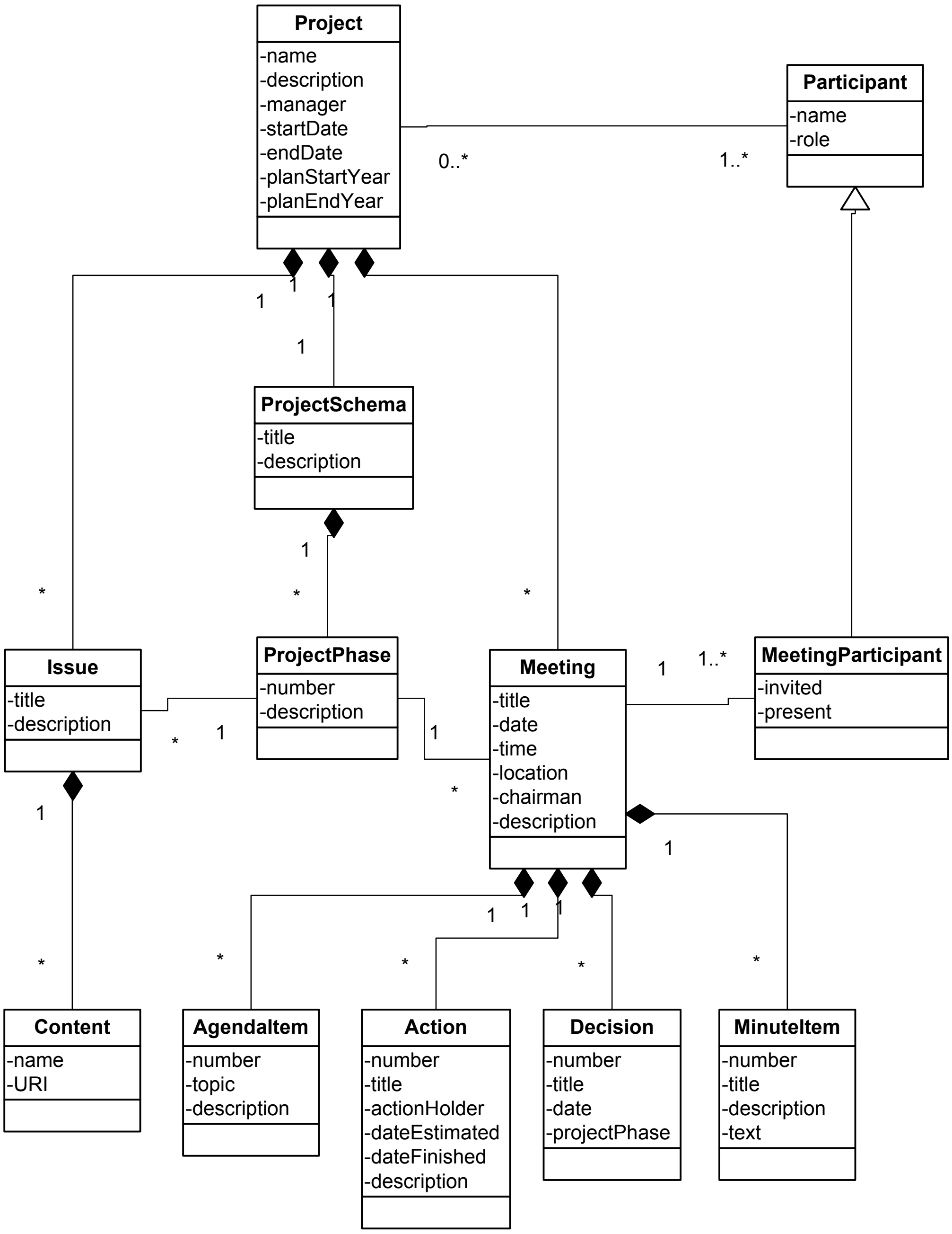


Figure 67: UML conceptual class diagram of the process support services.

The information classes that we described are made available as services in the suite. For example, Figure 68 shows a list of actions that can be provided as a portlet in a web portal. For each action at which meeting the action was specified, what the action is, who is the action holder, the deadline and the actual date of completion is displayed. The other services can be presented in a similar fashion.

Meeting	Nr.	title	action holder	estimated	finished
21-7-2015	1	Lezen opdrachtverstrekking gebiedsplan Maasvlakte 2015	-- allen --,	29-7-2015	29-7-2015
21-7-2015	2	Inlezen in gebiedsplan Noordwesthoek 2004	-- allen --,	29-7-2015	29-7-2015
21-7-2015	3	Inlezen in gebiedsplan Hartelstrook 2005	-- allen --,	29-7-2015	29-7-2015
21-7-2015	4	De beschikbaarheid van kavels op de Maasvlakte onderzoeken	Ristic, Linda	29-7-2015	29-7-2015
21-7-2015	5	Huidige bezetting in de Maasvlakte in kaart brengen	Olde Hanter, Bart-Luc	29-7-2015	29-7-2015
29-7-2015	6	Set van criteria, doelstellingen en taken (aspecten) herzien	-- allen --,	5-8-2015	5-8-2015
29-7-2015	7	Herzien en aandragen van invullingsvormen die moeten worden beschouwd	-- allen --,	5-8-2015	5-8-2015
5-8-2015	1	De toekomstige wegvervoervraag op de maasvlakte benaderen tot 2018	van der Hoop, Kees	12-8-2015	12-8-2015
5-8-2015	2	De geluidsbudgetten op de Maasvlakte bepalen	Lenger, Cleo	12-8-2015	12-8-2015
5-8-2015	3	Mogelijkheid om een spoorverbinding te realiseren	Pons, Cees	12-8-2015	12-8-2015
5-8-2015	4	Bodemverontreiniging op de Maasvlakte in kaart brengen	Lenger, Cleo	12-8-2015	19-8-2015
5-8-2015	5	Onderzoeken mogelijkheden verbetering nautische bereikbaarheid Lyondell optie	Sibbes, Rob	12-8-2015	
12-8-2015	1	Risicocontouren van rapportageplichtige bedrijven beschikbaar maken	Lenger, Cleo	19-8-2015	
12-8-2015	2	Knelpuntenanalyse doen van de wegbereikbaarheid tussen 2015 en 2018	Pons, Cees	10-10-2015	
12-8-2015	3	Aanwezigheid van zeldzame soorten controleren	de Groot, Jasna	19-8-2015	
12-8-2015	4	Marktvraag naar havengebied binnen de marktsegmenten bepalen	Olde Hanter, Bart-Luc	19-8-2015	
12-8-2015	5	Geplande beheersprojecten op de Maasvlakte in de APS updaten	de Groot, Jasna	19-8-2015	
19-8-2015	1		van der Hoop, Kees		
19-8-2015	2	Uitstootgegevens bulkbedrijven	van der Hoop, Kees	1-10-2015	
19-8-2015	3	Opvragen leidingengegevens	Ristic, Linda	10-10-2015	
19-8-2015	4	Overzicht van geintresseerde bedrijven	van der Hoop, Kees	10-10-2015	
19-8-2015	5	Mogelijkheid uitstel groot onderhoud	de Groot, Jasna		

Figure 68: example of a list of actions.

6.7 Conclusions

In this chapter we described the services of the mainport planning suite. We followed Papazoglou and Georgakopoulos (2003) in that a service provided by an IT system is defined as a self-describing and open component that supports the rapid, low-cost composition of a distributed information system. We concluded that the concept of a container is an essential element in the design of mainport planning services. We focused specifically on the services that support the planning activities, such as analysis, design, evaluation, and choice, called respectively *Map*, *Sketchbook*, *Matchbox,* and *Aspect Explorer*. Furthermore we introduced services that support the planning process, such services to set agendas and agree on actions.

This chapter concludes the design of our MPS. With the design of the building blocks and the services of a MPS, in respectively chapters 5 and 6, we fulfilled the requirements which we presented in chapter 4.

- Requirement 1 (usefulness) was fulfilled with the design of Map, Sketchbook, Matchbox and Aspect Explorer in 6.2 – 6.5

- Requirement 2 (usefulness) was fulfilled with the design of the visualization library in 5.4
- Requirement 3 (usefulness) was fulfilled with the design of the map library in 5.5
- Requirement 4 (usefulness) was fulfilled with the division of our suite in building blocks in 5.3
- Requirement 5 (usefulness) was fulfilled with the design of the services that support the process of mainport planning in 6.6
- Requirement 6 (usability) was addressed in the GUI design of our MPS services in 6.2-6.6
- Requirements 7, 8 and 9 (usability) were was addressed in the design of the portlet event library 5.6
- Requirement 10 (usage) was fulfilled with the choice for a programming platform 5.2
- Requirement 11 (usage) was addressed by our choice for a portaled suite 5.2

We should note that we did not explicitly address requirement 6, the usability of the GUI, because the focus of this research is not centered on the design of user friendly GUIs. However, we had to address this requirement implicitly because the quality of our MPS GUIs could influence the outcome of its evaluation. Furthermore requirement 11 has thus far only been addressed by introducing a technical solution, i.e. the portlet API, which supports the usage of a MPS based on role-based authentication schemes which follow organizational policies. In the following chapters we test our design by evaluating its prototype implementations.

7 Evaluation of APS version 1

In chapter 2 we introduced an inductive case study in the Port of Rotterdam (PoR). We explored which needs should be addressed by a suite of services to support area planners in the port. We translated these needs to the requirements for our suite and presented the design of our suite in chapters 5 and 6. In this chapter we describe how we tested our MPS in an evaluation session at the Port of Rotterdam. During this evaluation session, a number of experienced area planners were guided through a realistic studio-based area planning meeting, which was supported by our MPS. The objective of this evaluation session was to evaluate to what extent our MPS meets the needs that we identified in chapter 2.

First we introduce the evaluation in 7.1. Next we present the design of the APS1 prototype. In 7.3 we present the structure of the evaluation. In section 7.4 we describe the outcomes of the evaluation session from which we draw conclusions in section 7.5.

7.1 Introduction

We chose to evaluate our design using a prototype of our suite. We adhere to Crinnion (1991) who states that a prototype is a physical working model of the proposed system, which can be used to identify weaknesses in our understanding of the “real” requirements. We wanted the evaluation of the prototype to be realistic, meaning that the setting in which the prototype was to be tested should be as close to reality as possible. Furthermore, we wanted to do the evaluation at the PoR because of the strong involvement and support of the people in this organization. We discussed the possibility of using the prototype in a real area planning project at the PoR, however, this option was rejected by the PoR representatives because it could put too much stress on the continuity of the area planning project. The area planners in the PoR were willing to help and participate in the evaluation of our suite, but they would not use the suite in a real project before it was extensively tested, and integrated with the organizational and IT infrastructure. Instead, our prototype was used in an evaluation session: a fictive but realistic area planning meeting that represented studio-based planning in the PoR. We invited a multi-disciplinary team of experienced area planners of the PoR for the evaluation session.

The area planners recognized the need to develop further studio based port planning and where willing to participate actively in the design and evaluation of our suite. On the one hand we are grateful for this opportunity; on the other hand we must be aware of a possible bias in the outcomes of the test. We describe the details of the evaluation session in the following sections.

7.2 Description of the APS1 prototype

We implemented a prototype of the MPS which was named the Area Planning Studio version 1, or APS1. A detailed description of the prototype can be found in (Chin et al. 2006b). The implemented services of the suite were described in detail in chapter 6. The prototype did not feature the full set of designed services. The services provided were: the set of designed planning process services (action lists, meeting notes etc.), *Map,*and *Matchbox*.

In chapter 5 we introduced the design of the suite’s building blocks, and in chapter 6 we assembled these building blocks to form the design of the suite’s services. To implement the

APS1 prototype we needed to make a number of choices regarding the deployment of the suite:

- Which web server should we use?
- Which servlet container should we use?
- Which portal should use?

In chapter 5 we chose Java EE extended with the portlet specification as our programming platform. Java EE specifies the interfaces for servlet containers, but it does not provide us with a servlet container implementation. To get an implementation of a servlet container we have to choose from vendor specific products. Similarly, the portlet specification provides a library of interfaces and a reference implementation, but not a working portal solution. We chose Apache Tomcat as a combined webserver and servlet container solution, and we chose Apache Jetspeed as our portal solution (Figure 69). These choices were interrelated because the Apache Software Foundation provided a pre-configured bundle of Jetspeed and Tomcat. Our choice was mainly a choice for Jetspeed because it offered a free, stable and open-source implementation of the JSR168 portlet standard. Furthermore, Jetspeed offered an extensive set of tools for user, role, and group management, as well as a fully configurable page navigation structure.

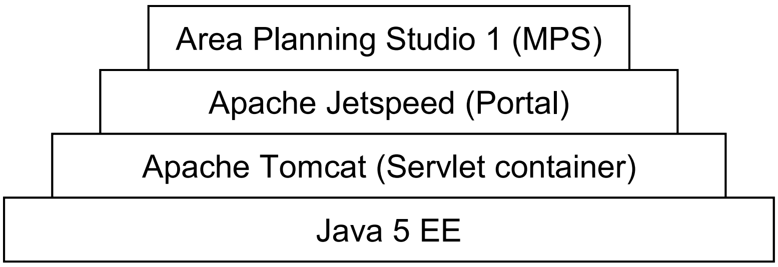


Figure 69: APS1 deployed in an Apache Jetspeed portal, which runs inside the Apache Tomcat server.

Following a classification for web applications as stated by Genender et al. (2006), our choice resembles a lightweight solution, because it implies that APS1 runs on a single server that can serve multiple clients. This lightweight solution was sufficient to evaluate APS1 during our evaluation session; i.e. we did not have to deploy APS1 on a cluster of distributed servers. This configuration is shown in Figure 70. The APS1 services were made available as portlets, which were aggregated and visualized as portal pages by Jetspeed. Some portlets include applets to generate rich user interfaces such as interactive maps. Furthermore APS1 uses a mySQL database as a persistency service.

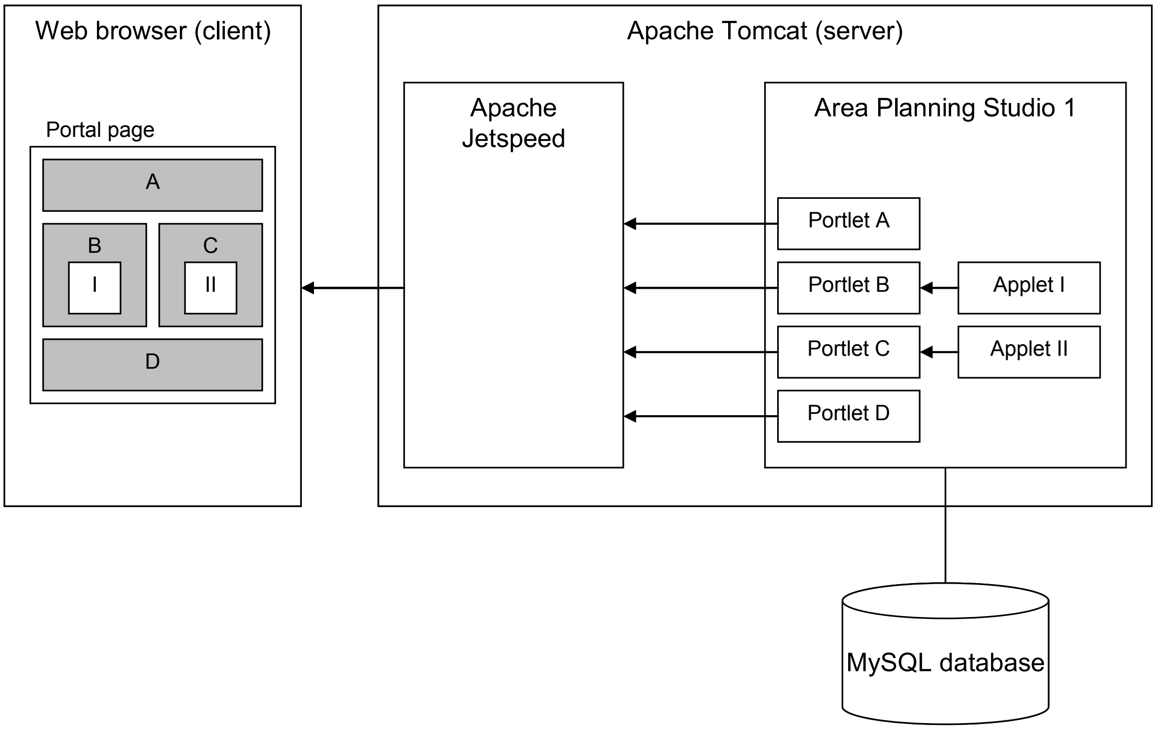


Figure 70: APS1 deployment.

The services provided by the APS1 prototype were structured on portal pages by configuring Jetspeed's navigation structure (Figure 71). This navigation structure provides a horizontal set of tab-buttons. Each tab-button leads to a different section, which can be subdivided into a vertical column of menu-buttons. Each button is used to display a different portal page. Figure 72 shows the relation between the services offered by the APS1 prototype. At the bottom of the image is *Maintenance* which represents portlets for managing user profiles, roles and project templates. This data is input to Project Administration which are portlets to manage, create, edit, or delete, projects and to assign users and roles to projects. The portlets provided in *Project Administration* are assigned to the area planning project leader, while the *Maintenance* portlets are meant for the department that maintains APS.

The other services in APS1 are meant to be used by the area planners. *Project Information*, *Meeting Information* and *Issue Information* are portlets related to the process of planning such as action and decision lists. There is a hierarchical relation between the project portlets and meeting- and issue portlets. The project portlets refer to actions and decisions of an entire project, while the meeting- and issue portlets filter information related to respectively area planning meetings and issues. An issue is a specific topic which is considered during an area planning project. For example, ground pollution at a specified location can be an issue. This issue can, however, be discussed during several area planning meetings, possibly in combination with other issues. Hence, meetings and issues are two distinct ways of slicing through the available project information resulting in different views on the same information. In addition discussion forums were defined for each project.

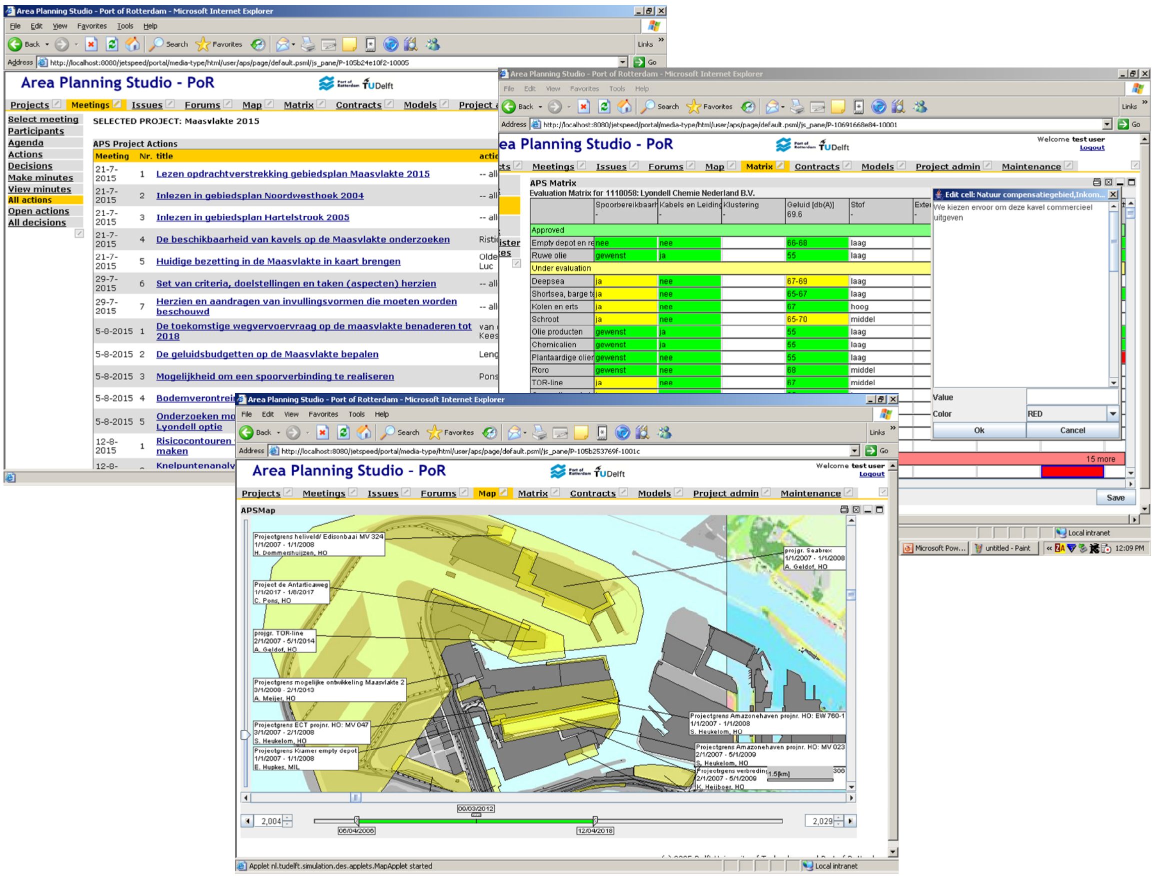


Figure 71: Screenshots of the APS1 prototype.

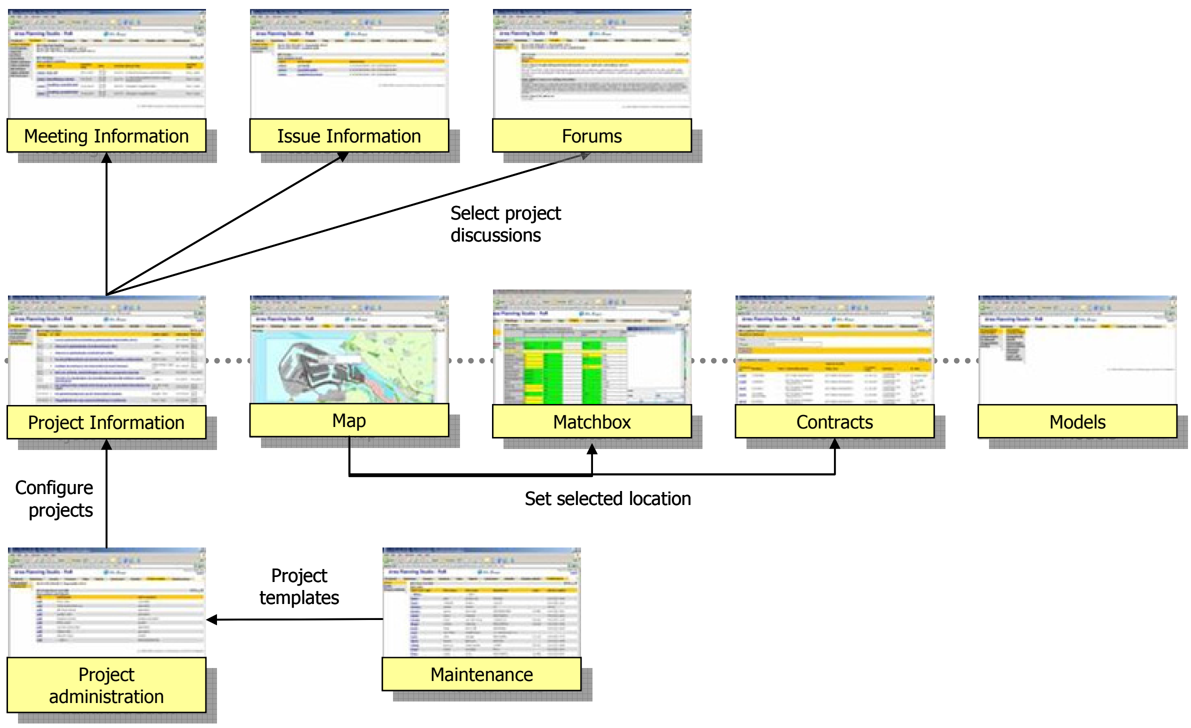


Figure 72: The structure of the APS1 prototype.

Map and *Matchbox* are services related to the content of area planning. *Map* and *Matchbox* were made available directly in the web browser as applets. When clicking on a location (a lot) in *Map* a label appears which displays the name of the lot and the current customer. The lot name propagates to *Matchbox* and the *Contracts* portlet, such that both can display information for the selected lot. *Contracts* is a portlet which resembles the port's customer database called COREN. The idea was to demonstrate that *Map* can link to distributed data available in the organization. In this prototype we created a copy of a subset of the COREN data to demonstrate the concept. Furthermore we added a *Models* portlet to demonstrate that in a later stage simulation models can be added to the suite.

7.3 Structure of the evaluation session

We chose to center the evaluation of APS1 on a single day at the headquarters of the PoR. The location, a meeting room, was chosen because it resembles the environment in which area planners are used to working. During discussions with experts in the port several alternatives for an evaluation session were examined. It was decided to set an area planning meeting based on a realistic but fictive planning situation. During this area planning meeting a team of area planners would be invited as to use APS1, and they would be guided through a storyboard that resembled common tasks in an area planning meeting, and at the same time they could be provided with the opportunity to use different functionalities of APS1. Apart from the participating area planners, an observation desk was arranged for academic researchers and employees of the PoR; and because of constraints on the availability of area planners and other employees of the PoR, it was decided to limit the session to roughly half a day, with time for discussion afterwards.

In the following sections we introduce the details of the evaluation session. We describe the selection of the participants of the evaluation session in 7.3.1. We describe the storyboard and setup of the evaluation session in 7.3.2. We present our choice for the evaluation instruments in 7.3.3.

7.3.1 Selection of participants

The participants should be experienced area planners in the PoR. Similar to a real area planning meeting, they should play different roles during the meeting. This meant that we searched for participants from different departments in the port's organization such as the infrastructure department, the commercial department, the environmental department. Furthermore we wanted to involve the GIS analysts, because their role as providers of geovisualizations was considered to be crucial for the success APS1.

The formation of a team of participants was organized by the APS1 development team of the port, in close cooperation with the academic researchers. The APS1 development team, which was headed by the executive manager of the port's infrastructure department, represented the port throughout the preparations for the evaluation session. They had access to managers in other departments and therefore they could establish the necessary appointments.

Seven participants took part in the area planning team (Figure 73), these were:

- project leader port development, session chairman
- senior advisor operational management, especially nautical expertise

- senior business development manager logistics, especially concerning commercial attractiveness
- advisor, especially concerning environmental issues and infrastructure
- advisor environmental affairs, especially concerning environmental issues and special development
- advisor geo information
- management assistant

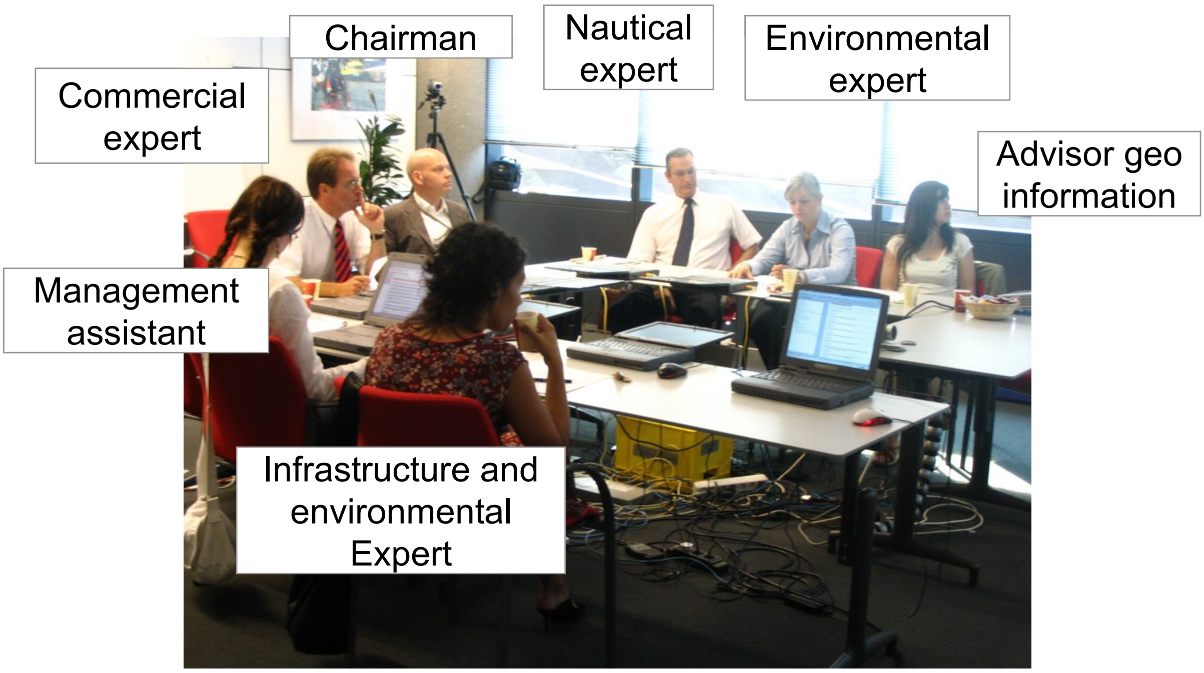


Figure 73: During the test session.

7.3.2 Setup of the evaluation session

The setup of the test room is shown in Figure 74. The room was set up using mobile Group Decision Room (GDR) equipment from the TU Delft, which was used to gather feedback and generate ideas for improvement after the session (see section 7.3.3). Furthermore a desk was made available for the APS1 servers: the computers that hosted the APS1 software. As we described in 7.2 we implemented APS1 on a single server. However, during the evaluation session we deployed APS1 on two separate servers: one was used to host the session and another one was used as a backup. Both servers were also used as clients to present information using two beamers. In addition, a laptop was used as an APS1 client computer for the management assistant to make meeting minutes, and record of actions and decisions in the *Project Information* portlets.

A storyboard was developed to guide the participants through the features offered by APS1 and to let them experience the ideas that were developed regarding studio based area planning. The storyboard was used to present a fictive situation in 2015 in the Maasvlakte area (see Appendix C). During the test session the participants played through an area planning meeting that was supposed to represent one of multiple meetings in a planning project. Hence, we prepared information concerning previous meetings as input for this meeting. In the storyboard issues were raised concerning a lot named the "Lyondell option" for which an analysis regarding e.g. nautical reachability, endangered species, commercial attractiveness, and geographic location should be made. The participants used the services provided by the APS1 prototype to make this analysis.

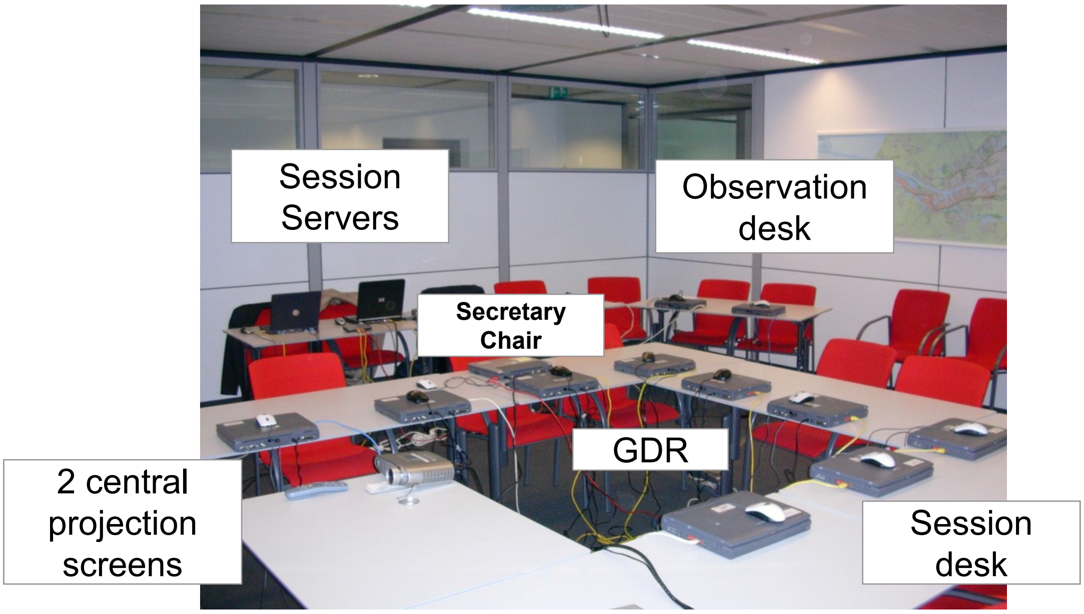


Figure 74: APS1 test room setup.

7.3.3 Evaluation instruments

We selected a number of evaluation instruments for the evaluation of APS1, which will be described in this section. Each test instrument sheds a different light on the evaluation. Our main question addresses the effectiveness of studio based mainport planning (chapter 1). APS1 should improve the effectiveness of area planning in the PoR. This improvement is relative to the current way of working. Therefore we needed two comparable measurements: the perceived effectiveness of the current way of area planning, and the perceived effectiveness of area planning using APS1.

Furthermore we needed a way to measure effectiveness. We broke down effectiveness into a number of sub-goals: situation awareness, perceived usefulness, perceived ease of use and satisfaction. These sub-goals each refer to an equivalent theory that we shall address in the following subsections. These theories also provided us with accepted questionnaires to do measurements. The questionnaires are formulated in the form of statements that can be answered on a Likert-scale (Jamieson 2004). We decided to develop two questionnaires: one to measure the perceived effectiveness of the current area planning processes, and one to measure the perceived effectiveness of using APS1 in area planning. The questionnaire that related to the current way of working was filled in before the test session, and the other questionnaire was filled in after the test session. The statements in both questionnaires were largely equivalent, which means a statement in the first questionnaire had an equivalent statement in the second questionnaire. This enabled us to make a comparison between the pre and post test measurements.

Apart from the use of questionnaires we chose to use a number of other measurement instruments. We chose observation as a means to get a more detailed insight into the activities that were carried out during the test session. Furthermore we used the mobile Group Decision Room (GDR) of the Delft University of Technology to collect feedback immediately after the evaluation session and to facilitate a brainstorming session about possible improvements for APS1 (Briggs and De Vreede 2001, Briggs et al. 2003). This system digitally processes the results and enables the participants to react to each others comments while they remain anonymous. After the GDR session, the participants could discuss their experiences with APS1. Finally in the week after the session, each participant was interviewed to gain insight into their personal opinion on APS1.

To summarize our selection of research instruments:

- questionnaires before and afterward the test with statements about situation awareness, perceived usefulness, perceived ease of use, and satisfaction.
- observation of the activities and their collaborative nature: Which activities are carried out by whom?
- GDR-session: collecting feedback about the first impression, and generating ideas to improve APS1.
- discussion of the experiences of the participants.
- open-ended interviews with individual participants in the week after the test session.

In the following subsections we will explain the details of respectively the questionnaires, observations, GDR-session, the discussion and open-ended interviews.

7.3.3.1 Questionnaires

We based the pre and post test questionnaires on three distinct theories. From these theories we formulated statements that could be answered on a seven point Liker scale (Jamieson 2004). As argued by Jamieson we can compare the pre and post test measurements using the Mann–Whitney U-test (Mann and Whitney 1947, Wilcoxon 1945). Furthermore we supplemented these statements with statements that were focused on the specified functionalities for our suite. Below we describe the theories used in detail and at the end of this section we provide the statements that were selected for the questionnaires.

Situation awareness
In chapter 2 we concluded that visualization services are an important aspect of a suite to support mainport planning. Furthermore, in chapter 3 we clarified the idea that information visualization is used “to amplify cognition” (Card et al. 1999), i.e. to reduce the cognitive load when working with an abundance of information. Often visualization is seen as an externalization of memory (Ware 2005). Situation Awareness (SA) is a concept that can be used to measure the effects mentioned before (Endsley 1995). The theory of SA assumes that if the cognitive load is lowered, people are better capable of overseeing the situation on three distinct levels. A better performance on all three levels will finally result in improved decision making. The three levels are:

- perception: people observe more elements
- comprehension: people are better capable of valuing the elements
- projection: people are better able to translate elements to the future

SA is influenced by a number of factors, such as training of people, capacity of supporting systems, workload and experience (Figure 75). SA can be measured using SART (Situation Awareness Rating Technique). In SART the participants individually give a score for thirteen different aspects. Together these aspects provide an indication of SA. The aspects are grouped in three categories: Demand, Supply and Understanding. The SA can be calculated using: SA = Understanding – (Demand – Supply) where:

- **Demand**: is demand on cognitive resources, instability of situations, complexity of situations, variability of situations
- **Supply**: is supply of cognitive resources, readiness, concentration of attention, division of attention, spare mental capacity
- **Understanding**: is understanding of the situation, information quantity, information quality, familiarity with the environment

In SART the participants indicate in how far they agree with a number of statements. Each of the statements refers to one of the thirteen aspects. It should be noted that SART was originally developed for individuals tasks. Since we used SART in a team session, we used the statements in the "we-form". Furthermore, to keep the total length of the questionnaire within acceptable limits we had to limit the amount of statements.

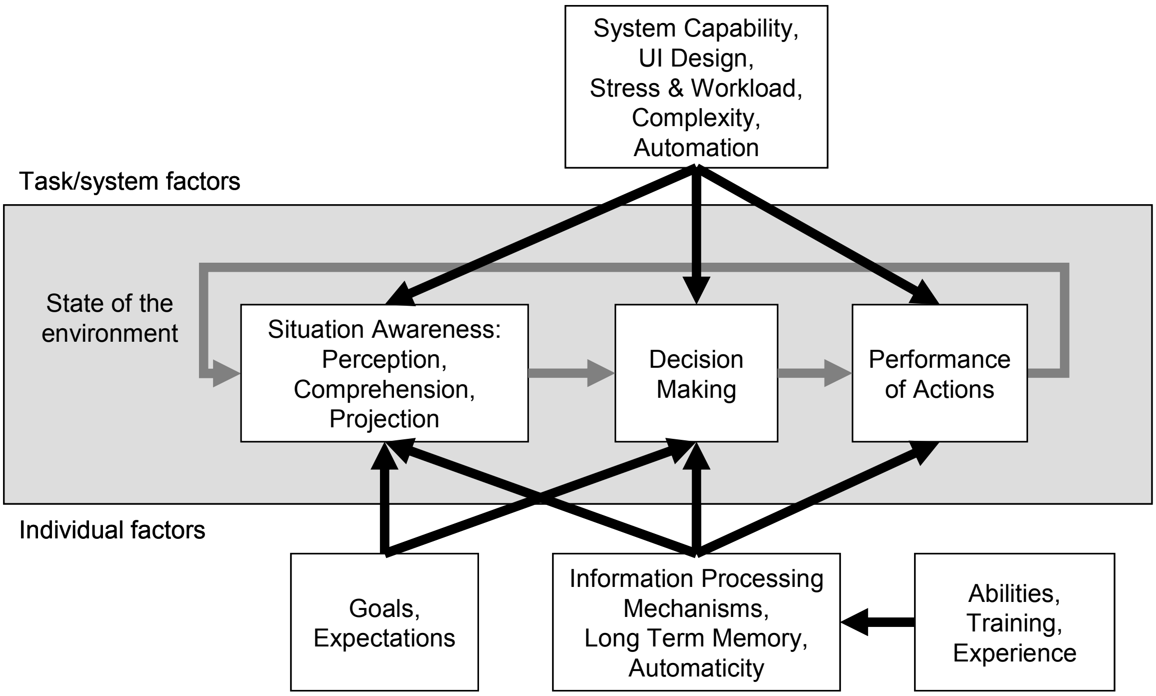


Figure 75: SART (Endsley 1995).

Technology Acceptance Model

The added value of an IT system only becomes truly visible when the system is actually used in practice. TAM (Technology Acceptance Model) (Davis 1989) is a useful instrument to measure to what extent a system will be used in the future. In TAM it is assumed that the decision to deploy a system in practice depends on two factors: perceived usefulness and perceived ease of use (Figure 76): What is the expected added value of the system? How easy to use do users expect the system to be?

For our suite, perceived usefulness was closely related to situational awareness. Just like SART, TAM measures on a five point scale using statements. While SART focuses mainly on the tasks which should be performed, TAM focuses strongly on the system meant to support these tasks. In that sense TAM can be regarded as an extension of SART. Statements related to perceived usefulness could only be answered after the APS1 had been demonstrated in the test session.

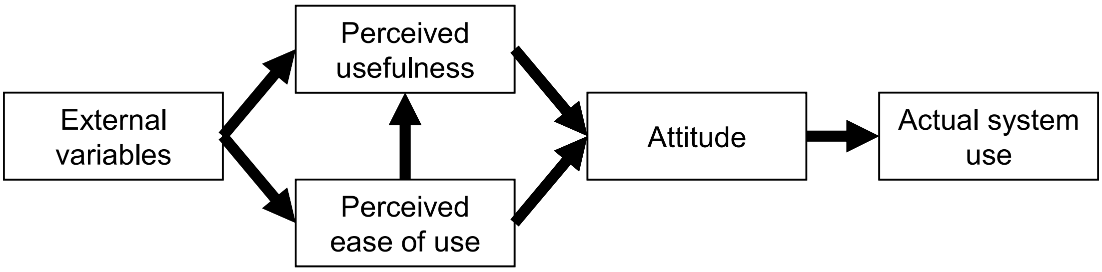


Figure 76: TAM (Davis 1998).

Perceived ease of use is shown in how far users think that they can use the system with ease. Similar to the other instruments this aspect is measured using a questionnaire. Perceived ease of use can only be evaluated after the participants actually used a system.

We choose to use TAM for testing the APS, however we had to take in to account a number of limitations. Firstly, TAM is meant to be used after a person has really used a system, but during our test session not all the participants would directly work with the APS. Consequently we decided to leave perceived ease of use out of the questionnaire. Secondly TAM was designed for individual work processes while we aim at a team of area planners. To deal with this limitation we decided to change the TAM questionnaire into the we-form; and finally, TAM is meant to evaluate a tool, but the APS1 is more than a tool as it consists of a suite of software services. We used TAM to evaluate APS as a whole, while during the discussion and the interviews we evaluated the individual services of APS1.

Satisfaction

Using SART and TAM we can already provide a substantial insight into the future use of the APS. Another aspect of importance that provides insight in the future use of the APS1 was user satisfaction with the outcome and the process. Research shows that when a system is considered useful and easy to use, it still is not guaranteed that the system will actually be used. Satisfaction plays an important role herein and depends on many factors. It is even possible that a system that is not considered to be very useful or easy to use will be used when the users are satisfied with the system for other reasons.

Therefore we chose SAT (Satisfaction Attainment Theory) (Bruce 2003) as a third instrument to measure the use of APS1. In SAT a questionnaire is designed to measure the satisfaction of users. SAT puts an emphasis on two levels: satisfaction with the outcome of the process and satisfaction with the process itself. During the test session the outcome was less important because we used an imaginary case. Therefore we only focused on satisfaction with the process, which was again measured using statements.

Statements used in the questionnaires

In the literature, predefined questionnaires with statements can be found regarding the described theories (e.g Davis 1989 and Bruce 2003). These questionnaires needed to be customized for our particular situation. Furthermore simply combining existing questionnaires would result in a list of statements that was too long to answer, regarding the time available for the participants. Therefore a selection of statements was made, which were further specified for our evaluation session. Furthermore statements were added to address specific functionalities of APS1. To ensure the quality of our questionnaire, an expert in decision support systems and evaluation sessions was consulted. Finally, the statements were specified as in Table 1.

Table 1: Statements used in the questionnaires.

Nr.	Pre and post test statement (translated from Dutch)	Theory used
1	**pre:** I feel satisfied about the way area planning is done. **post**: I feel satisfied about the way the evaluation session was held.	SAT
2	**pre:** Area planning process costs a lot of effort. **post**: The area planning process during the evaluation session costs a lot of effort.	SART
3	**pre:** I feel satisfied with the means used to support area planning **post**: APS1 supported the area planning process well.	SAT
4	**pre:** The area planning process is effective **post**: The area planning process during the evaluation session is more effective compared to the normal process.	SART
5	**pre:** The area planning process supports the realization of Port Plan 2020. **post**: The area planning process during the evaluation session supported the realization of the Port Plan 2020.	SAT
6	**pre:** The area planning process is efficient **post**: The area planning process during the evaluation session was more efficient compared to the normal process.	SAT
7	**pre:** The time used for area planning is well used **post**: The time for the area planning process during the evaluation session was well used.	SAT
8	**pre:** I have a lot of useful information at my disposal for the area planning process. **post**: A lot of useful information was placed at my disposal for the area planning process.	SART
9	**pre:** The information in an area plan is consistent **post**: APS1 presents the information in a consistent way.	TAM
10	**pre:** I can concentrate on the most important aspects of the area planning process and I was not distracted by irrelevant details. **post**: During the evaluation session I could concentrate on the most important aspects of the area planning process and I was not distracted by irrelevant details.	SART
11	**pre:** During the area planning process a lot of different aspects are simultaniously important, and I experience difficulties in focusing on one aspect at a time. **post**: During the area planning process of the evaluation session a lot of different aspects were simultaneously important, and I experienced difficulties in focusing on one aspect at a time.	SART
12	**pre:** During area planning an integral evaluation of alternative land uses for a specified lot is made.	-

	post: APS1 supports the integral evaluation of alternative land uses for a specified lot.	
13	**pre:** I have enough insight into the different spatial aspects during an area planning process. **post**: APS1 provides enough insight into the different spatial aspects during the area planning process.	-
14	**pre:** I have enough insight into the transition of an area from a current situation to a future situation. **post**: APS1 provides enough insight into the transition of an area from a current situation to a future situation.	-
15	**pre:** I have enough insight into the development of the area planning process. **post**: APS1 provides enough insight into the development of the area planning process.	-
16	**pre:** I have enough insight into the future effects of alternative land uses (e.g. regarding road traffic load). **post**: APS1 provides enough insight into the future effects of alternative land uses, e.g. regarding road traffic load.	-
17	**pre:** The retrieval of choices made during the area planning process is well supported. **post**: APS1 sufficiently supports the retrieval of choices made during the area planning process.	-
18	**pre:** In general I have a good overview of an area planning process. **post**: In general I had a good overview of the area planning process.	SART
19	**pre:** I find it easy to communicate with team members during an area planning process. **post**: APS1 makes it easy to communicate with team members during the area planning process.	-
20	**pre:** I find it easy to manage the tasks of team members during the area planning process. **post**: APS1 makes it easy to manage the tasks of team members during the area planning process.	-
21	**pre:** The different stakes of stakeholders from the port were evenly treated during an area planning process. **post**: The different stakes of stakeholders from the port were evenly treated during the area planning process of the evaluation session.	-
22	**pre:** An area plan supports the needs of its users such that an area is developed with vision. **post**: An area plan supports the needs of its users such that an area is developed with vision.	-
23	**pre:** It is possible easily and quickly to maintain an area plan when changes occur in the port or at potential customers. **post**: APS1 makes it possible easily and quickly to maintain an area plan when changes occur in the port or at potential customers.	-

In addition to these questions the post measurement contained questions regarding the perceived usefulness of APS1 (Table 2). These questions were based on TAM (Davis 1989) and did not have a pre measurement counterpart, because no system equivalent to APS1 is used in the current PoR planning process.

Table 2: Additional statements regarding the perceived usefulness of APS1.

Nr.	Statements regarding the perceived usefulness of APS1
24	APS1 presents information in a structured way
25	APS1 presents relevant information
26	APS1 presents information at the right level of detail
27	I felt confident when working with APS1
28	Using APS1 requires a lot of mental effort
29	I was often confused when using APS1
30	APS1 is flexible it its usage
31	Using APS1 is cumbersome to use

7.3.3.2 Observation

The previously mentioned instruments were all used to measure the perceptions with respect to APS1 of the participants after the session. To put these perceptions into perspective a number of employees of the PoR and researchers from the Delft University of Technology observed the session from the observation desk. Furthermore a video was made of the entire evaluation session. Examples of observations that can be made are listed below.

- Errors in APS1 usage such as pressing the wrong button, wanting to press a button which is not available. These are errors only a user can make.
- Misinterpreting the APS1 such as using another functionality than the intended one. These are errors that can also be made by non-users, such as the observers.
- Determining which tasks are performed by individuals and which by the team as a whole.
- Determining how often, and when, the focus of the participants shifts away from the area planning process.
- Determining which parts of APS1 are used for what, this depends on the scenario used during the session.
- Determining when users need assistance in using APS1.

7.3.3.3 GDR-session

The instruments mentioned thus far were focussed mainly on individual measurements to gain insight into the added value of the system without immediately providing insight into directions for improvements. The GDR provided an electronic environment to facilitate brainstorming about possible improvements for APS1. Further, we used the GDR to collect feedback regarding the first impression of the participants.

The GDR-session was organized in three parts:

1. collecting feedback with respect to the demonstrated functionalities and planning process,
2. identifying important (additional) functionalities for APS, and
3. identifying bottlenecks for the introduction of APS1 in the port organization.

To collect feedback with respect to the demonstrated functionalities and planning process, we made use of the "Topic Commenter" of the GDR (Briggs and De Vreede 2001). The Topic Commenter is a tool by which session participants can anonymously react to a list of topics. They can also see each other's (anonymous) reactions and react to these as well. The topics concerned the added value of: the portal, process support services, *Matchbox* service, and *Map* service. Furthermore, we formulated topics regarding APS1s support for individual tasks, support for users in different roles, and support for different phases in an area planning project. Additionally a topic was added to collect "any other comments".

To identify important (additional) functionalities and bottlenecks for the introduction of APS1 in the port organization, we made use of the "Categorizer" of the GDR (Briggs and De Vreede 2001). This tool facilitates brainstorming about subjects and the categorization of similar ideas.

7.3.4 Discussion and open-ended interviews

The discussion and open-ended interviews were not structured in detail. We recorded the discussion on video, and the interviews were recorded using a voice recorder. To analyze the discussion and the open-ended interviews we followed the methods used for qualitative research described by Creswell (2003) and Baarda et al. (2005):

- to organize and prepare the data for analysis
- to obtain a general sense of the information
- to label the information to create meaningful "chunks"
- to categorize and organize the labeled information
- to describe the findings
- to interpret the findings and answer the research question

7.3.5 Summary of the prepared evaluation session

Two teams cooperated in the planning of the evaluation session: a team of researchers from the university, and a team of managers and experts from the PoR. The team of researchers from the university focused on the implementation of the APS1 prototype and the evaluation instruments. The team of the PoR was responsible for providing feedback from a customer viewpoint. Furthermore they made the project known throughout the departments of the PoR, established appointments with relevant experts, provided content for the evaluation session such as maps and reports, and arranged the necessary facilities such as a room for the evaluation session.

We planned the evaluation session as follows:

- the development team implemented an APS1 prototype, prepared fictive planning content, and developed a storyboard for the test session
- in the week before the session all participants received a questionnaire to evaluate the current area planning process
- on the day of the session the participants were guided through a storyboard, where they were introduced in studio based area planning using the APS1 prototype

- immediately after the test session feedback was gathered on the first impressions and ideas for improvement using the mobile GDR (Group Decision Room) of the Delft University of Technology
- the test day ended with an open discussion concerning the experiences and lessons learned by the participants
- in the week after the session all participants received a questionnaire to evaluate the APS
- open ended interviews were planned with the individual participants to obtain detailed feedback

7.4 Outcomes of the evaluation session

We discuss the outcomes of the evaluation of APS1 in terms of the different measurement instruments. We start with the outcomes of the questionnaires followed by the observations made during the session, the outcomes of the brainstorming session and the discussion directly after the session. Finally we describe the outcomes of the interviews with the individual participants in the week after the test session.

7.4.1 The outcomes of the questionnaires

Table 3 and Figure 77 provide an overview of the results from the pre and post measurements i.e. respectively the answers of the questionnaire before and after the session. The participants chose among seven discrete answers, where 1 means totally disagree with the provided statement, while 7 means totally agree with the provided statement. Table 3 shows the statements (in an abbreviated form) and the outcomes for the pre and post test measurements. The Mann-Whitney U test is a non-parametric test to determine whether the medians of two samples of observations are the same. A Mann-Whitney U-test was applied on the outcomes of each statement using the commonly used probability of 95% (Mann and Whitney 1947, Wilcoxon 1945, Law and Kelton 2000). From this test we can distinguish those statements for which a significant improvement was found. Our null hypothesis and alternative hypothesis are defined as:

$$H_0 : \mu_2 \le \mu_1 \tag{1}$$

$$H_1 : \mu_2 > \mu_1 \tag{2}$$

Where:
μ_1 is the median of the pre test measurement statement
μ_2 is the median of the post test measurement statement

Statements 2 and 11 (see Table 1) were formulated negatively, in which case we swapped the indices 1 and 2.

The U values were computed using the following steps.

- for each statement we called the pre measurement sample1 and the post measurement sample2, when we expected the pre measurement to be lower on average compared to the post test measurement and vice versa
- we subsequently took each value in sample1 and counted the number of values in sample2 that were smaller than it, we counted half for equal values

- the result of the count was the U value

This method is referred to as the "direct method" to compute the value of U. Alternatively the "indirect method" can be used:

$$U = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1 \qquad (3)$$

where:
n_1 is the number of values in sample1
n_2 is the number of values in sample2
R_1 is the sum of ranks, i.e. relative positions of values, in sample1.

The resulting U-value is compared with the critical U-value for 95% probability for one-tailed testing. The critical U-values can be found in a U-table (Mann and Whitney 1947). If U < U95% then the difference between the pre and post test measurement is considered to be significant with a 95% probability and then we can reject H_0.

The bar chart shows the geometric means of the answers of the participants on the seven point scale. We are mainly interested in the difference in value between the pre and post test measurements. The bar chart clearly displays the outcomes shown in Table 3: regarding all questions the situation after the session was more positive than before the session, or that no significant difference was found. From these answers we can draw some first indications which are listed below.

- The participants expect the area planning process to be less difficult when using APS1
- The participants found that with APS1 they were provided with a better means to support area planning
- The participants expect area planning to be more effective and efficient with APS1
- The participants expect to use their time better when using APS1
- The participants expect to be better capable of keeping the information consistent when using APS1
- The participants expect to be better capable of doing area planning in a more integral way when using APS1
- APS1 was expected to improve the insight in the development of the area planning process
- APS1 was expected to help in retrieving the choices made during an area planning process
- The tasks of team members could be better managed when using APS
- APS1 was expected to enable a quicker maintenance of area plans
- No significant improvement was found in e.g. the visualization of the transition of an area towards the future and in providing insight into the future effects; note: we will use the interviews and discussion outcomes to address these topics in more detail later

Table 3: Outcomes of the pre and post measurements.

Abbriviated statement	1	2	3	4	5	6	7	1	2	3	4	5	6	7	G1	G2	d=G2-G1	n1	n2	U	U95%	Significant	
I feel satisfied with how the area planning process progresses		1		3	2	1			1		1	2	2		4.09	4.39	0.3	7	6	16.0	8	No	
The area planning process is difficult				3	2	2			4		1	1			4.79	2.62	-2.17	7	6	5.5	8	Yes	Negative statement
The means support the area planning process well		2	1	1	2							4	2		3.26	5.31	2.05	6	6	4.0	7	Yes	
The area planning process is effective		2	2	1	1						2	1	3		2.99	5.08	2.09	6	6	3.5	7	Yes	
Area planning supports the realization of the port master plan	1	1		2	1	1			2			3	1		3.14	3.8	0.66	6	6	14.0	7	No	
The area planning process is efficient		3	2	1						2		1	3		2.57	4.62	2.05	6	6	4.0	7	Yes	
The time for area planning is well used		1	1	2	1	1						2	4		3.77	5.64	1.87	6	6	5.0	7	Yes	
I got a lot of useful information to my availability in area planning				2	2	2						1	3	1	4.93	5.97	1.04	6	5	6.0	5	No	
The information in an area plan is consistent	1	1		1	3							3	2		3.16	5.38	2.22	6	5	4.5	5	Yes	
I can concentrate on the most important aspects in area planning and I am not distracted by details	1	1	2	1		1			1	1		1	2		2.75	4.04	1.29	6	5	9.5	5	No	
In area planing a lot of aspects are important at the same time, I experience difficulties in focussing on one aspect at a time	1	2	2		1			1	3			1			2.38	2.09	-0.29	6	5	12.0	5	No	Negative Statement
In area planning all aspects are evaluated in an integral way			1	3	2						1	1	2	1	4.11	5.5	1.39	6	5	4.5	5	Yes	
I have enough insight in the spatial aspects of area planning		1		2	1	1	1		1		1	1	2		4.34	4.28	-0.06	6	5	15.0	5	No	
I have enough insight in the transition of an area from the current situation to the future		1		2	3						1	1	3		3.98	5.33	1.35	6	5	5.5	5	No	
I have enough insight in the development of an area plan process		1		3	1	1						2	2	1	3.96	5.75	1.79	6	5	4.0	5	Yes	
I have enough insight in the future effects of alternatives land uses (e.g. regarding road traffic load)		2	2	1	1					1	3	1			2.99	4.95	1.96	6	5	8.0	5	No	
Choices made during the area lanning process can easily be retreived	2		1	1	2							1	2	2	2.59	6.15	3.56	6	5	1.0	5	Yes	
In general I have a good overview of the area planning process				4	1	1					1	2	2		4.44	5.14	0.7	6	5	8.0	5	No	
I find it easy to communicate with team members during area planning				1	2	2					1	2	1	1	5.14	5.3	0.16	5	5	11.5	4	No	
The tasks of team members are well managed during area planning		1	1	1	2								2	3	3.59	6.58	2.99	5	5	0.0	4	Yes	
The different stakes of the port are evently treated		1		2	2				1			3	1		3.81	4.32	0.51	5	5	7.5	4	No	
An area plan supports the needs of its users such that an area is developed with vision	1		1	1	2					1	1		3		3.13	4.82	1.69	5	5	6.0	4	No	
An area plan can easily and quickly be maintained when changes occur in the port or potential customers	1		1	2	1							3	2		2.99	5.38	2.39	5	5	1.5	4	Yes	

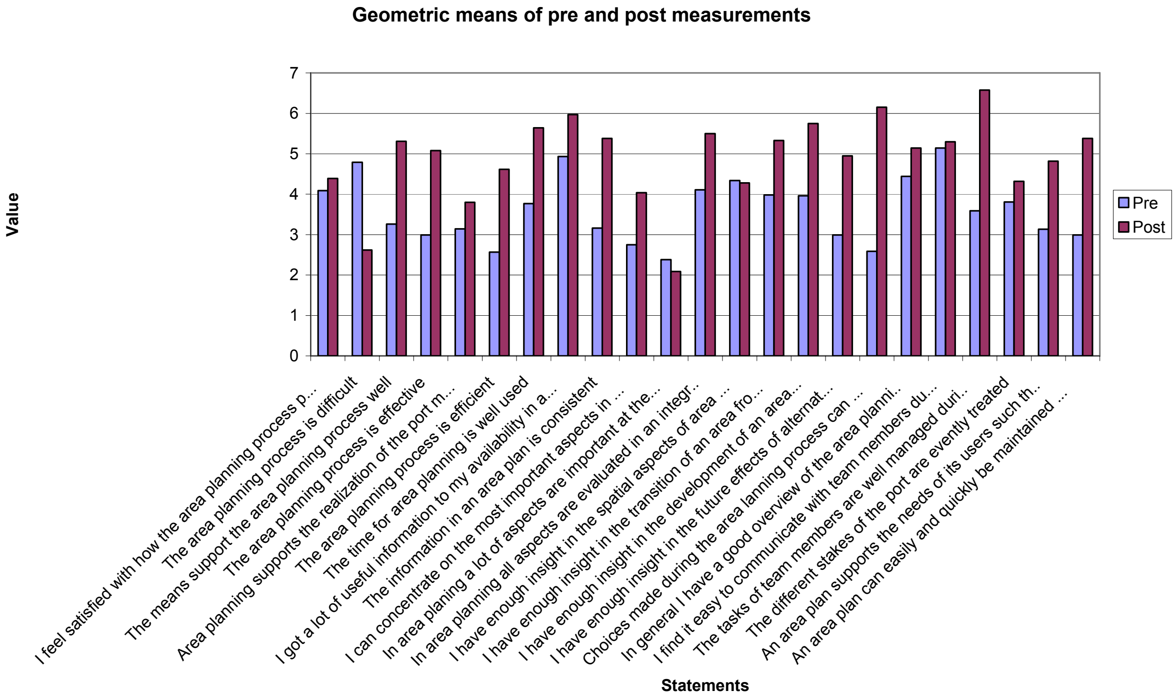


Figure 77: An overview of the difference between the pre and post measurements (geometric means).

We also applied a Wilcoxon Signed Rank Test on the geometric means G1 and G2 of all statement pairs using SPSS statistics software. The outcomes are given in Table 4 and Table 5. The significance value for 1-tailed testing is less than 0.0005, hence we can conclude that the two sets of geometric means are significally different. (Pallant 2005)

Table 4: SPSS ranks output

Ranks

		N	Mean Rank	Sum of Ranks
G2 - G1	Negative Ranks	1[a]	1.00	1.00
	Positive Ranks	22[b]	12.50	275.00
	Ties	0[c]		
	Total	23		

a. G2 < G1

b. G2 > G1

c. G2 = G1

Table 5: SPSS test statistics output.

Test Statistics[b]

	G2 - G1
Z	-4.167[a]
Asymp. Sig. (2-tailed)	.000
Exact Sig. (2-tailed)	.000
Exact Sig. (1-tailed)	.000
Point Probability	.000

a. Based on negative ranks.

b. Wilcoxon Signed Ranks Test

The outcomes regarding the perceived usefulness of APS1 are presented in Table 6. Further the geometric means of the statements about the perceived usefulness of APS1 are shown in Figure 78. These outcomes cannot be related to the current area planning process, because there is no comparable suite in the current way of working. We see that especially the way in which the information was presented by APS1 is a positive point. Furthermore the participants felt confident using the APS, but at the same time they indicate that using APS1 was combersome.

Table 6: Outcomes regarding the perceived usefulness of APS1.

Statement	1	2	3	4	5	6	7	n
APS presents the information in an structured way					2	3		5
APS presents relevant information					2	3		5
APS presents information at the right level of detail			1	2		2		5
Confident when working with APS					2	2	1	5
Using APS requires a lot of mental effort		3		1	1			5
Often confused when using APS		2		3				5
APS is flexible in its usage			1	1	2	1		5
Using APS is cumbersome		1	1		1	2		5

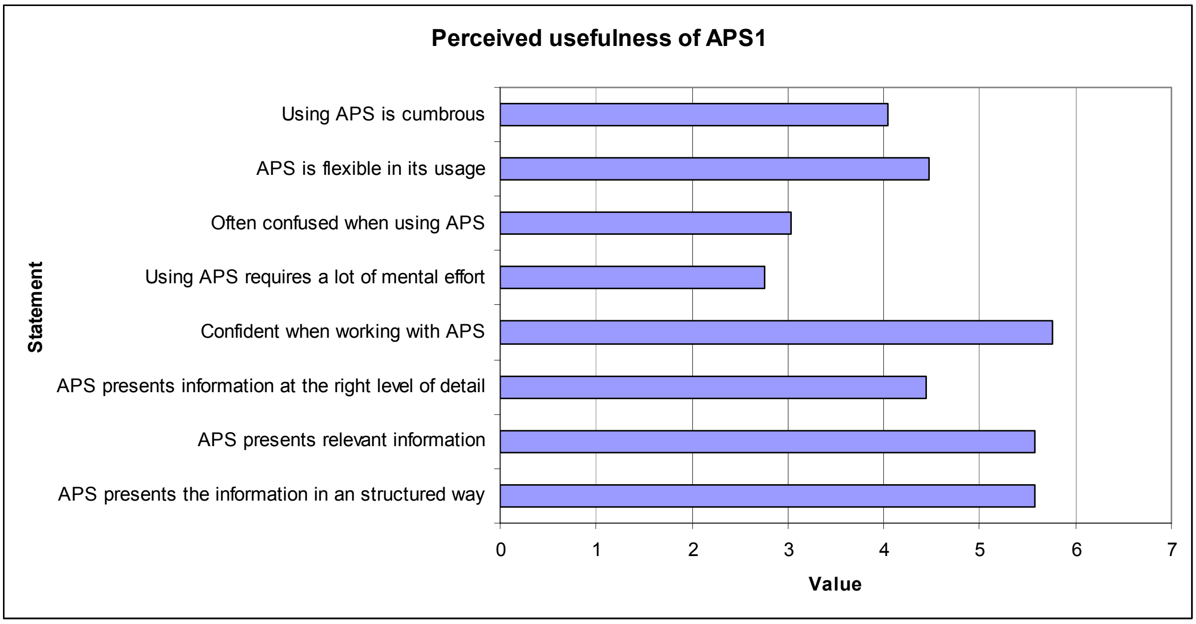


Figure 78: Perceived usefulness of APS1 (geometric means).

The participants also made textual remarks in the questionnaires regarding the current way of area planning. Below we present a summary of these remarks.

- Currently area planners do not make full use of the GIS systems available in the PoR. They still rely on paper maps which makes it difficult to look up information. Improved support for GIS is considered to be useful.
- Area planning is a long term process, while in the meantime the area planners also participate in other projects. This makes it difficult to stay focused and to stay involved. The process takes too long, leading to thick piles of paper. The process should be better structured.
- Most of the information required for area planning is available, but it is not always complete or up to date. Data from electronic databases still has to be linked manualy.
- Area plans contain too many details, are too little customer oriented and should be edited more effectively. In addition it is not always clear which choices were made or why certain choices were made.
- Area plans are currently not related to the strategic Port Plan 2020. In practice area plans become obsolete quickly because customers are given contracts, and inconsistencies at the boundaries with neighboring areas force the plan to be altered. Consequently there is too little consistency between different area plans. Furthermore area plans are static documents and do not take into account the dynamics of the environment and the transition to the future situation.
- Currently commercial affairs do not use area plans because these do not answer their needs.

7.4.2 Observations during the test session

From observations made during the APS1 test session we saw that:

- the management assistant, who used the APS1 directly to enter meeting minutes, actions and decisions, had some difficulties with the user interface which in some situations worked counter intuitively
- some participants were distracted by the large amount of technology in the room, which included the GDR laptops and network cables
- drawing directly on *Map* was considered more as a gadget than a useful tool (see the next section)

7.4.3 GDR-session and discussion after the session

The GDR was used to collect feedback with respect to the demonstrated functionalities and planning process. Furthermore, the GDR session was aimed at extracting a number of ideas for improving the APS1, and identifying bottlenecks for introducing an MPS in the port organization. The participants started using the GDR as was intended, but soon they indicated that they were overloaded with ideas from the session. Therefore they did not see the added value of brainstorming about additional features or improvements. Yet, the discussion round that followed revealed some interesting ideas that should be considered for further development. Consequently the GDR session and the discussion that followed cannot be considered separately, and therefore we will discuss the outcomes of these two instruments together. Further, we have structured the outcomes according to the three U's as defined by

Keen and Sol (2007): usefulness, usability and usage. These three U's are largely equivalent to the perceived usefulness, perceived ease of use and attitude towards usage as defined in TAM (Davis 1998).

Usefulness

The APS1 should lead to quicker and thinner area planning reports. The participants agreed that the APS1 can improve the effectiveness and efficiency of area planning by supporting a more dynamic planning process in which plans are updated in a rolling manner. The APS1 is expected to leave more room for creative thinking, because the time required for editing and administrative tasks can be reduced.

A digital map is in itself nothing new, but in an area planning process *Map* has added value. The current RIV system, the port's GIS viewer, is considered to be too complicated and with a focus on displaying the current situation. A digital map as demonstrated during the test session is easier to use, and can also be used to display alternative designs for an area in the future. Functionalities that could be added are tools to measure distances on the map, drawing tools to visualize distances and ranges, the use of templates to draw e.g. scale models of ships, and the possibility to print a map; and finally, it was mentioned that *Map* could also be used to search for location based content, such as photos made at different locations in the port.

As a part of *Map* we demonstrated the possibility to make digital sketches instead of sketching design ideas on paper. The advantage is that digital sketches can immediately be processed and stored in APS1. However the participants did not consider digital sketching convenient. One of the most experienced area planners who participated in the test session, especially was rather skeptical about digital sketching. He found that he could not express his ideas as well as he was used to doing using paper.

The *Matchbox* was considered to be a useful means for visualizing the status of an evaluation of alternative land uses. *Matchbox* makes it very explicit how alternatives are evaluated and why. Furthermore *Matchbox* acts as a checklist to see which aspects still need to be considered or require more attention. However, the participants also indicated that updating the data in *Matchbox* requires a considerable amount of discipline.

In terms of content, the APS1 should also provide legal scopes and development plans. Furthermore the APS1 could be used to display a customer prospect list of the commercial department. A distinction between information about the current situation, a "base case", and alternative designs for an area should be made in the APS1.

It was mentioned that the APS should provide functionality for freezing a plan in a certain status. Planning is a dynamic process while reality changes as a plan is developed, freezing different versions of a plan, allows a history of the planning process to be recorded at regular intervals. This way a log, or blog is created of the area planning process. A final area plan should still be made that represents a frozen stage of the latest available information. Whenever the situation changes the latest area plan can be updated.

We aimed to demonstrate a road traffic simulation model as part of APS. However, due to time restrictions this part of the test session was skipped. During the discussion we came back to this concept. In general, the participants did not directly see the added value of developing such a model. Their main concern was that simulation models should not become separate systems that end up somewhere on a shelf. For example the port is currently developing a road traffic model in cooperation with the Ministry of Transport. Furthermore different simulation models should be linked to gain a better insight into and understanding of the relations between different aspects, such as NOx emissions and road traffic.

Usability
The technology was considered too distracting. One of the participants mentioned that the APS is like “a Christmas tree full of balls”. This was also one of the reasons why the participants did not feel like brainstorming about additional functionalities. Or as they said: “Please, no more balls in the tree”. Instead of adding additional functionalities, the APS should be simplified considerably to increase its ease of use. The participants also felt that the APS was focused too much on process, i.e. decisions, actions, agendas, and too little on the content, i.e. analysis, design, evaluation.

Furthermore the participants had difficulties in focusing on the content, firstly because of the share amount of computers and wiring in the room (including the GDR) and secondly because they were not used to watching images that are projected on the wall during area planning meetings. In other words, the environment that we created for the test did not resemble the kind of room in which they were used to working. Consequently some participants indicated that they felt uncomfortable in this artificial setting. This considerably influenced their perception regarding the ease of use of the APS1.

Usage
Probably the most predominant remarks made during the discussion were about the actual usage of APS. The term “authorization” was mentioned regularly: Who will be authorized to operate the APS? Who maintains the APS? Who is responsible for the correctness of the data? The participants agreed that the implementation of APS1 in their area planning process would substantially change their way of working. As a result, they needed to develop clear policies regarding APS1 usage. Further research is required to determine these policies.

Finally, area planning can be considered from the perspective of the available land, or from the perspective of the customer. A proper balance should be found between the two extremes. When considering an area only from the perspective of the available lots an area plan may be developed that does not fit actual customer demand. A pure customer perspective better fits the needs of the commercial department, but will fragment the port industrial region.

7.4.4 Interviews

The outcomes of the interviews follow the lines of the discussion after the test session as reported in 7.4.3. The interviews revealed that the participants favored further development of the APS. The APS1 provided a useful platform for keeping track of how a planning project develops over time; served to update area plans at a regular basis, i.e. in a more “rolling manner”, leading to thinner, more to the point area plans; and to integrate information available within the organization. However, the usability of the suite needed to be improved, especially regarding the *Matchbox* service and the overall structure of the portal. The APS1 prototype put too much emphasis on process, e.g. decisions, actions, minutes, which caused some of the participants to feel distracted regarding content. Some participants also feared that using the APS1 would require too much discipline. The participants saw that the APS1 provides good opportunities for a better information access, especially regarding geographic maps; and the *Matchbox* service was considered to be a valuable checklist when evaluating customers for a specified lot.

Further some ideas for added functionalities were expressed such as a tool to measure distances on the map. In addition the comment was made that the commercial department should be fully involved. Finally, and possibly most importantly, concerns were raised regarding the actual usage of the suite in a real planning process: the organizational and technical requirements for actually deploying the APS1 needed to be worked out further. Who is going to maintain the system and authorize its users? The participants stressed that to actually start using the APS1 there should be clear policies on user rights and information management.

7.5 Conclusions

Based on our findings in the evaluation session we can draw a number of conclusions.

- Both from the questionnaires and the discussions with the participants we found that APS1 is perceived as a useful suite to support area planning.
- APS1 makes the information flow in port planning more explicit, resulting in a better awareness of the available information. The participants felt that their time was well spent when using the APS1. Using a suite such as APS1 is expected to lead to a more effective and efficient area planning process.
- APS1 enables a more dynamic form of area planning. While area planning currently focuses on creating a static image of the future situation, using an APS helps the planners to consider the transition to the future situation. Further use of an APS is expected to lead to thinner area planning reports which are more to the point. These reports can be kept up to date in a rolling manner as the situation in the port changes.
- *Map* and *Matchbox* were considered to be useful tools to support area planning. Map was seen as an accessible viewer for geographic information, which is, in contrast to RIV, also capable of displaying dynamic information. *Matchbox* was considered to be a visualization tool which provides insight into the status of an evaluation process for the involved area planners.
- APS should have a stronger focus on content and less on process. The APS prototype put too much focus on managing actions, recording decisions, making meeting minutes and so on. The participants found that they actually want to focus more on the content of area planning i.e. analyzing the area, designing alternative land uses, evaluating potential customers and industry types for specific locations, and eventually fix the choices made.
- Usability should be improved. The APS1 prototype was overwhelming in terms of the amount of features that were used during the test session. The participants found that using APS1 was cumbersome. Therefore, the ease of use of APS1 should be improved.
- Paper design sketches cannot be replaced by digital ones. We demonstrated the possibility of making digital sketches directly in APS1, but found that the participants did not like this idea. They felt that they could better express their creative ideas on paper and saw digital sketching as a gadget.
- Apart from usage in area planning, the participants mentioned that APS1 may also be useful in other projects. Especially from the standpoint of the commercial department, APS1 is expected to be useful in customer negotiations. Instead of bringing a paper area

plan to a customer, an employee of the commercial department can take a laptop and use it to help them to answer questions from a customer by directly accessing APS1. Information that specifically addresses the interests of the customer can be prepared in APS1 in advance.

- APS1 maintenance is a critical factor. The question was asked as to who was authorized to do what in APS1 and who is responsible to whom in the end. This is an area that clearly requires further research to develop appropriate policies.
- APS1 usage requires discipline. The participants mentioned that APS1 required them to work in a more disciplined way. The participants made it clear that if the area planners did not manage to keep the information in APS1 up to date, the whole system would become useless.

We conclude that overall the evaluation of APS1 was a success, but further research is required to improve the ease of use of APS1 and to determine policies for its actual usage in the organization (Chin and Verbraeck 2006a, Chin et al. 2006b, Smits et al. 2005, Chin et al. 2005b). In the next chapter we present the evaluation of APS version 2, which should address some of the shortcomings.

8 Evaluation of APS version 2

In this chapter we evaluate an improved prototype of the suite, the Area Planning Studio version 2. First we motivate our choice for this evaluation in 8.1. Then we present the structure of the evaluation in 8.2, in section 8.3 we describe the improved prototype. The outcomes of the evaluation are given in section 8.4 from which we draw conclusions in section 8.5. In the next chapter we shall use these conclusions and the conclusions of chapter 7 to answer the research questions.

8.1 Introduction

In the previous chapter we discussed a first evaluation of our suite, the Area Planning Studio at the Port of Rotterdam. The outcomes were promising, however during the test session a number of weaknesses and potential threats came to light (see chapter 7). Furthermore the APS prototype lacked some of the services that we described in chapter 6. Therefore we could only evaluate a part of our design.

Consequently we decided to develop the APS prototype further in a final iteration. APS version 2 (APS2) represents a prototype of the full suite as we designed it in chapter 6. Furthermore we took the comments of the first evaluation session as input for APS2: APS2 was expected to have increased usability, and a better balance between process and content support.

In this chapter we shall refer to the APS prototype of the previous evaluation session as APS1, whereas the prototype of this evaluation is named APS2. When no version number is given we refer to APS in general.

8.2 Description of the APS2 prototype

While the first APS1 prototype represented a subset of the services that we described in chapter 6, APS2 represents the complete set of designed services (Chin and Verbraeck 2006a, Chin et al. 2006b). We decided to change the underlying portal from the Apache Jetspeed Portal to the Liferay Portal, because we found the latter to be more user friendly and easier to configure (Figure 79). Furthermore, changing the portal proves that APS can be deployed on any JSR168 compliant portal without any changes needed in the source code.

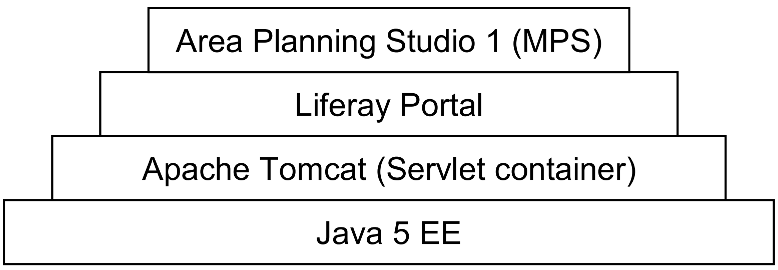


Figure 79: APS2 deployed in Liferay Portal.

In contrast to APS1 we did not use applets in APS2 (Figure 80). Instead we used Java's Web Start technology to make applications such as interactive maps available to the client. This has several advantages. One, a user can now easily open multiple application windows whereas applet instantiation is always restricted to the portlet in which it is displayed. This is, for example, useful when a user wants to open multiple maps and put them side by side for comparison. Additionaly a single portlet can now instantiate multiple web start applications, which is especially useful when opening content from *Sketchbook*. In this case *Sketchbook* can now provide access to the content, and open it in an appropriate viewer or editor. This does not exclude the use of applets, but in APS2 we found using Web Start applications to be more appropriate.

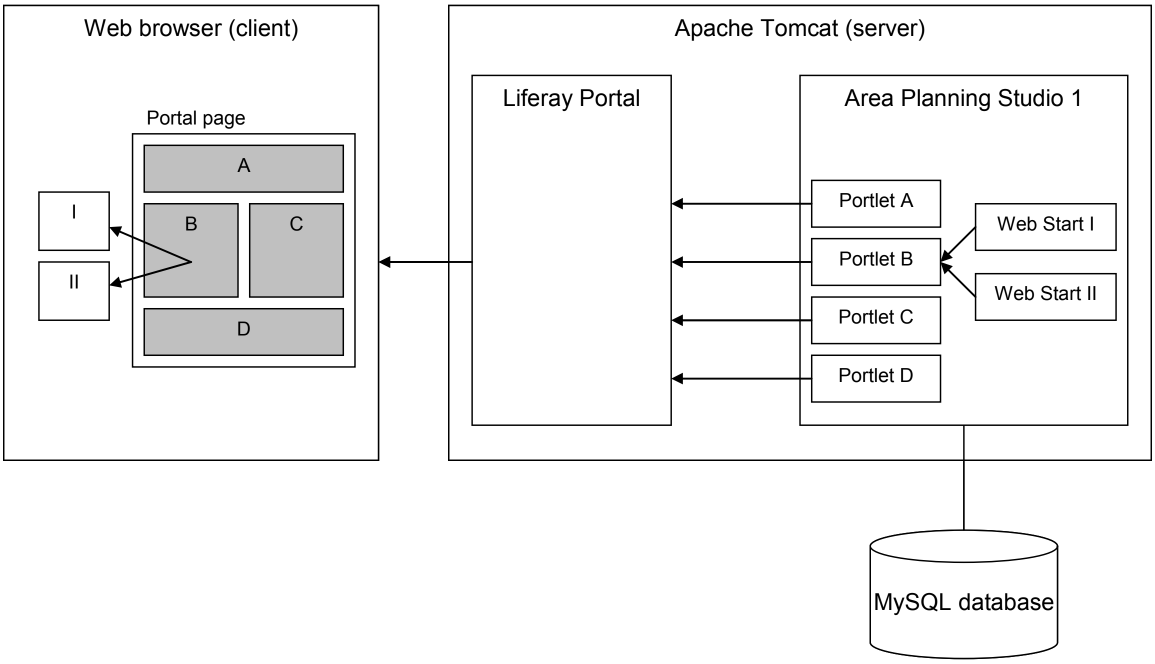


Figure 80: APS2 deployment.

In APS2 we changed the portal's navigation structure compared to the previous version to put more focus on the content related services instead of the process related services. *Sketchbook*, which was not available in APS1, was introduced in APS2 and became a central point of access to all content related information, such as maps, sketches, area photos and evaluations (Figure 81, Figure 82). The information in Sketchbook is either downloaded to the client or opened in a webstart application. While in APS1 Map and Matchbox were presented as applets inside the browser in APS2, both are now webstart applications that visualize data in Sketchbook. The advantages are twofold: different types of information can easily be put next to each other on the screen, and e.g. multiple maps can be opened simultaniously which was not possible in APS1.

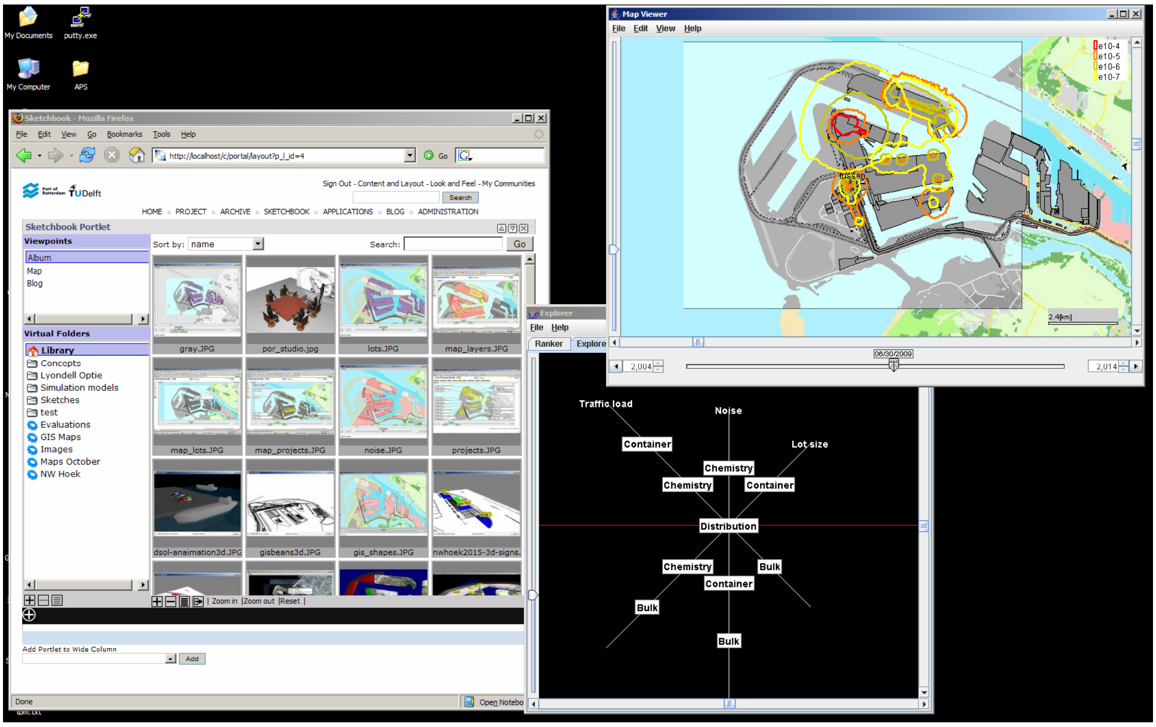


Figure 81: APS2 screenshot.

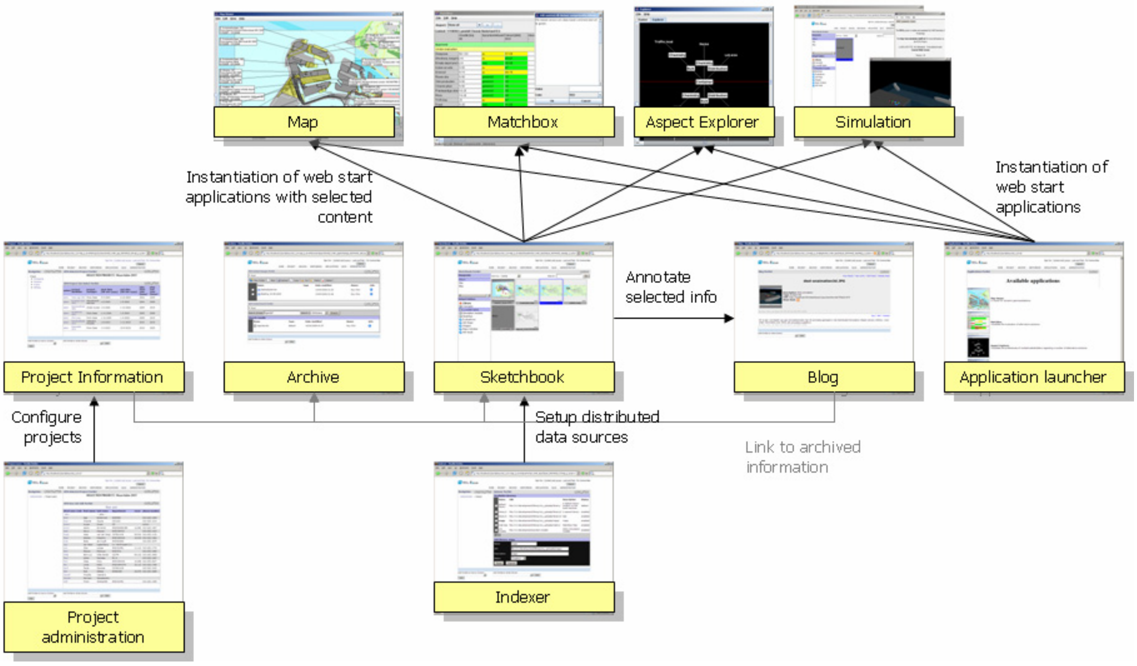


Figure 82: The structure of APS2.

8.3 Structure of the evaluation session

The aim of the evaluation of APS2 was twofold:

1. to evaluate the services which were not evaluated in the previous session
2. to determine whether APS2 (as a whole) is indeed an improvement over APS1

In addition we had the opportunity to meet with experts who could give feedback on how APS2 related to the state of art in their field i.e. give their opinion regarding the **innovativeness** of our solution. This expert feedback helps us to ensure that we are not reinventing an already existing solution and it helps us to gain additional insight into the added value of our design.

Figure 83 provides an overview of the evaluation. The evaluation was structured in a number of sessions at the Port of Rotterdam. The details of these sessions are explained in the following sections.
Before holding the sessions at the Port of Rotterdam we also had the opportunity to present APS2 to the Environmental Department of the port. During this meeting, which involved around 20 people, we gave a presentation on the APS and demonstrated the software, followed by a discussion. Most of the audience had heard of the APS but was not involved in its development. Furthermore, we were able to interview an expert in Planning Support Systems (PSS) (Brail and Klosterman 2001, Geertman 2002, Vonk et al. 2005). This expert was unfamiliar with our research and can therefore be considered to be independent.
These meetings provided a different angle on the evaluation of APS2 and were used as a check on the outcomes of the sessions held at the port. When the outcomes of the sessions are in line with remarks made during the discussion held with members of the Environmental Department at PoR, or during the interview with the independent expert, this diminishes the chance of a possible bias in the outcome of the sessions.

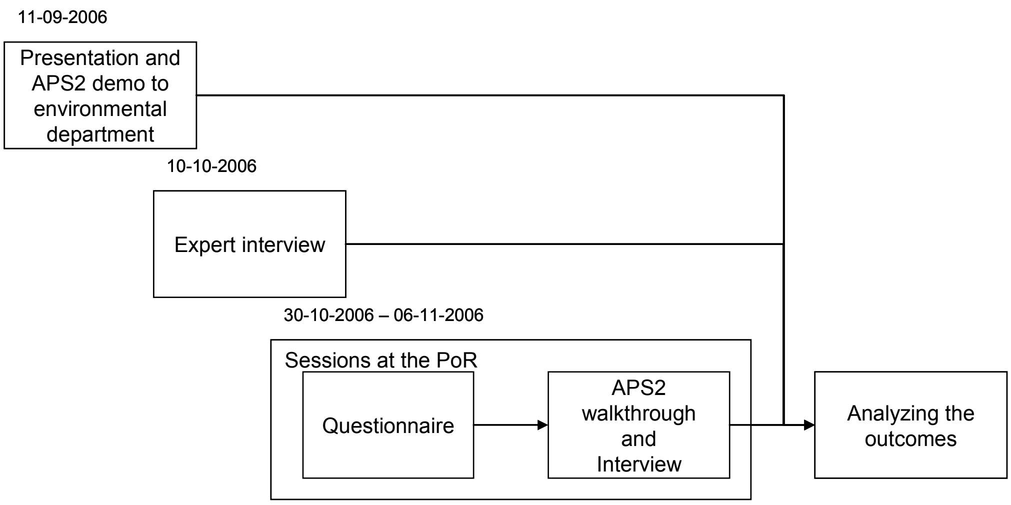


Figure 83: Structure of the evaluation.

8.3.1 Selection of participants

In line with the previous evaluation session, the evaluation of APS2 took place at the PoR, and should have involved the same set of participants. However, after contacting the port it turned out that not all the participants of the previous session were available. Out of the seven participants of the first session three were available for a second session. In addition three other participants could be arranged. These participants were already closely involved in the project and two of them were present as observers during the first session. Hence all participants were well informed about the project and actively participated in the previous phases.

It was difficult to bring all the participants together for a meeting, therefore a single session with a representative team of area planners was not an option. Instead a number of sessions were arranged, which gave us the opportunity to group participants based on their role in area planning projects. We organized three sessions: two sessions with participants from the infrastructure department and one session with participants from the drawing department. The participants from the drawing department were mainly involved in area planning as providers and managers of geographic information. Therefore they were considered to be the facilitator of APS2, while the other participants could be considered to be direct end-users of APS2.

8.3.2 The attitude of the participants before the session

Before having a session a reference point was needed. This evaluation session came more than a year after the previous session which raised a number of questions.

- How well could the participants still remember the previous session?
- Had their attitude towards APS change during the last year?
- Did they still support the outcomes of last year e.g. did they still consider APS to be useful?

To capture the starting point we developed a questionnaire (see Appendix D). In this questionnaire we first asked if the participants could still clearly remember the previous session. Next we used four open questions to ask the participants for their attitude towards APS, and finally, we used statements to gain insight into their support for the outcomes of the previous session. These statements were based on the questionnaire used in the previous session and could be answered using a five-point scale (Likert scale).

The outcomes of the questionnaire were used as input for the session i.e. to direct the interviews.

8.3.3 During the session

Each session was scheduled for two hours at the PoR headquarters. During the session a short presentation was used to refresh the minds of the participants. The presentation consisted of a recapitulation of the session of last year and the outcomes of that session. Furthermore the aim of the current session was clarified. Next a structured walkthrough of the APS2 prototype was provided by the researcher. We choose to use a walkthrough instead of an interactive storyboard, because the participants had already familiarized themselves with APS during the previous evaluation session (chapter 7), and therefore, we also did not have to introduce the participants to studio based decision making. The walkthrough focused mainly on the improvements made to the APS regarding usability and the new services. The data used in APS2 were the same as those used during the previous session i.e. the same storyboard was represented in APS2.
After the structured walkthrough a half open interview with the participants was conducted. This interview was structured along a number of axis:

- whether APS2 is indeed perceived as an improvement over APS version 1
- the perceived usefulness, perceived usability and attitude towards usage of the suite as a whole

- the usefulness of individual services provided by the suite, especially focusing on the newly introduced services: *Sketchbook* and *Aspect Explorer*

Matchbox and *Map* had already been evaluated in APS1 and were therefore not extensively addressed. Note: *Map* was addressed from a more technical side during the session with the drawing department.

In addition the participants of the drawing department were asked to rate the **innovativeness** of the GIS-related functionalities in the suite. During the PSS expert interview we also asked them to rate the innovativeness of the APS as a whole, including *Sketchbook* and *Aspect Explorer*.

8.3.4 Analyzing the session

To analyze the session we followed the methods used for qualitative research described by Creswell (2003) and Baarda et al. (2005), see section 7.3.4. The interviews were recorded on a voice-recorder. Each of the interviews was written out after the session and a transcription of the interview was sent to the participants for approval and possible corrections. Next, the transcripts of the interviews were labeled and categorized according to the axis identified in 8.3.3. In section 8.4, we describe the findings followed by an interpretation of the findings in section 8.5. The research questions will be answered in the next chapter.

8.4 Evaluation of APS2

We will discuss the outcomes of the evaluation of APS2 in terms of the axis identified in 8.4.3., and we will structure the outcomes along the 3U's (usefulness, usability and usage) and where relevant we will also address remarks regarding the innovativeness of the suite. In each of these topics we address the services to which the outcomes apply.
First we describe the outcomes of the discussion with the environmental department. Next we present the PSS expert opinion, followed by the outcomes of the evaluation sessions held at the PoR. Finally we interpret and compare the outcomes to eliminate possible biases and find similarities.

8.4.1 Outcomes of the meeting with the Environmental Department

Usefulness
Questions were asked about how exactly the APS will support the area planning process, i.e. where is the added value? Is area planning expected to proceed faster? Is it really an advantage when area plans are digitally available? In the discussion that followed the participants came to the conclusion that the APS changes the way of thinking about area planning. The APS makes very visible what information is available, is missing, is up-to-date; how information is inter-related and who owns the information. While previously problems that resulted from "information unawareness" were easily overlooked or simply ignored, the APS forces area planners to deal with information and available resources in a more conscious way. Furthermore the question was raised by the participants as to whether it would really help to digitally record the entire planning process: "Is it really useful to digitize the sketches of John which he would otherwise throw away? Nobody ever looks at those sketches again, do they?" Here again, the participants agreed that APS would lead to a new way of thinking about their planning process. Currently it is difficult to trace back why certain decisions were made, but with APS it becomes possible to go back along a digital timeline

and recapture the planning process. Finally the participants mentioned that the commercial department could use APS as an instrument in their negotiation with customers.

Usability
Ease of use was not discussed during this meeting.

Usage
Closely related to the new way of thinking about area planning, some organizational implications where brought up: Who will be responsible for maintaining APS? Who will be responsible for updating the data? Who can access what, etc.? New policies need to be developed and introduced in the port organization to deal with these issues. Note: issues related to responsibilities are also relevant without an APS, however, the absence of a support system in the current way of working makes it less crucial to find answers to these issues.

Innovativeness
It was mentioned that in urban planning the use of planning support systems is already more common. In port planning there are currently no support systems like APS as far as the participants could tell. There are "3rd part systems" such as those of the "kadaster". It would be useful to link to these systems instead of recreating the systems.

8.4.2 Outcomes of the PSS expert interview

Below we present the outcomes of the PSS expert interview.

Usefulness
The added value of APS, especially, comes to light in its ability to act as a kind of collective memory [of the planning organization]. Meeting minutes alone provide only a limited insight into a planning process. APS provides a much richer image that better fits the purpose of forming a vision. For example, APS makes it possible to trace back through time why certain design decisions were made. Furthermore the flow of information from meeting to meeting becomes more transparent.
Regarding the individual services the interactive timeline in *Map* was mentioned as a useful functionality for displaying dynamic data. Linking to (distributed) information from the map is also a strong point.
The added value of *Sketchbook* is mainly to be found in the possibility to put very different types of information together in a single medium. Designers do this very often in their search for alternative solution. For example, combining cyclo photos (360-degree photos) with design ideas has proven to be very useful. Therefore Sketchbook potentially can be very useful.
Regarding *Matchbox* and *Aspect Explorer* it was mentioned that it was a good choice to design services which visualize the subjective standpoints of the involved actors. In other PSS systems GIS is often directly coupled to multi-criteria analysis which does not really meet the needs of planners in practice.

Usability
Usability was not discussed during this interview.

Usage
It was mentioned that there is a tension between the need for a different way of planning on the one hand, and the possible negative consequences (need for authentication, discipline, visibility of social and political considerations) on the other hand.

Innovativeness
Importantly, APS can be considered to be a PSS, APS is a suite of services, “a toolbox”, and one can always wonder about the completeness of a toolbox. A GIS in itself is not considered a PSS because it is not dedicated to e.g. area planning.
In comparison with existing PSS, APS has similarities with SchetsGIS which was developed by TNO in the 1990s. However in APS different choices were made: where SchetsGIS allows planners to design directly on a digital map, in APS it was consciously decided not to allow this.
Regarding the individual services the interactive timeline in *Map* and *Aspect Explorer* especially were mentioned as innovative.

8.4.3 Outcomes of the sessions at the PoR

Below we describe the outcomes of the three evaluation sessions held in October 2006 and November 2006 at the PoR.

Questionnaire
Of the six participants, five returned the questionnaire. The participants indicated that they could still remember the previous evaluation session well, except for one participant who did not attend the session.

As the most important characteristics of APS they mentioned:

- recording actions and decisions during the planning process
- support for an integral approach to area planning, especially regarding *Matchbox*
- visualization of the area planning process and how an area develops over time

As positive conclusions of the previous session they mentioned:

- a strong potential in area planning and wide support of the participants
- the support APS offers for having an integral view on the available information
- the expectation that APS will make area planning more efficient

As negative conclusions of the previous session they mentioned:

- low usability and strong technology focus
- the risk of not keeping the data in APS up to date

In general all participants had a positive attitude towards the perceived usefulness of APS. Their opinion regarding APS had not changed significantly since the last session (see Table 7), also their opinion regarding the current process was unchanged: it takes a considerable amount of time, information is collected on an ad-hoc basis, and there is a need for a different approach to area planning. Furthermore they think that APS would be also applicable in projects which are not directly related to area planning.

Table 7: Outcomes of the questionnaire regarding the statements.

Question	1	2	3	4	5	?	Total	G
2.3 I expect that area planning will be more efficient with APS than without APS			1	3	1		5	3.9
2.4 I expect that area planning will be more effective with APS than without APS			1	3	1		5	3.9
2.5 I expect that APS will make it easier to retreive the choices that were made than without APS				2	3		5	4.6
2.6 I expect that using APS tasks can be better managed than without APS				3	2		5	4.4
2.7 I expect that using APS makes it easier and faster to maintain area plans than without APS				3	1	1	4	4.2
3.1 APS is useful for presenting information correctly and consistently			2	3			5	3.6
3.2 APS is useful for making integral evaluations of various aspects			1	2	2		5	4.1
3.3 APS is useful for making the transition of an area towards the future more transparant			1	1	3		5	4.3
3.4 APS is useful for getting insight into the process of area planning			1	2	2		5	4.1
3.5 APS is useful for addressing the various aspects in a more balenced way			2	1	1	1	4	3.7

Usefulness

The APS2 was indeed perceived to be an improvement over the previous version. Sketches, images and other types of information can now be combined, organized and compared from multiple points of view. In general the individual services offered by APS were considered to be useful and easier to use (see usability) as compared to the previous version. Furthermore the new portal navigation structure was considered to be an improvement.

The usefulness of APS2 is not to be found in e.g. the absolute time required for planning. Instead, its usefulness appears in facilitating the area planner’s ability to make the planning process more explicit and transparent. This means that it is easier to focus on the issues that really matter, without the danger of ending up in discussions that are already closed. Furthermore it becomes less important to arrange simultanious meeting of all participants, since people can now work more asynchronously at different locations.

One participant raised a question that was also raised during the previous session: that of whether APS should be a production system or a system in which area plans can be processed (an interactive project database). As a production system APS supports the design activities within an area planning project, while as an interactive project database it can be used to quickly scan for locations for customers. The APS as a project database is considered to be more useful because it can help managers to find a match between customer demands and available space in the port. However, this requires that all the relevant data about the customer and the area are available and up to date. APS can ideally be considered as a memory in which the process of area planning and its outcomes are stored. This memory can then be queried for relevant information. For example, when there is a customer who asks for a certain surface, a specific type of quay wall, etc. Then APS should be able to process this request to find locations which are suitable this customer. Subsequently these locations can be further queried for detailed information, such as soil data, to make a definitive evaluation for a specific customer. APS2 was not developed as such an interactive project database. Instead its services are focused on supporting and recording the planning process as it happens. The idea is a two stage rocket: first support the planning process itself, next use the accumulated information to use APS as an interactive project database. The second stage was defined by the PoR as a “vision”, a vision of APS in a later stage of development.

Another participant mentioned that it is more realistic to let a project team prepare the required information for a specific customer request. Thereafter the commercial department, supported by the prepared information, can start customer negotiations. “A representative of the commercial department can take a laptop to the customer to search for a location based on the prepared data.”

Map was considered useful especially because of its interactive timeline. Furthermore it was mentioned that maps can be prepared from model outputs such as the I^2-model (the PoR's noise model) by experts. When experts update these maps at regular intervals area planners can always access up-to-date information via APS.

Sketchbook was considered to be a useful service for putting together different types of information. This information can be prepared by the individual experts and made available during area planning meetings such that "one can quickly show something on the beamer". The virtual folders were considered to be a useful way to organize the information. The map viewpoint was considered to be specifically useful to relate photos of the port to location and time of creation. However, the participants also feared that the availability of the same data at multiple locations in APS could lead to confusion. Furthermore, one participant of the drawing department mentioned that Sketchbook should have more extensive support for comparing information.

Matchbox had been extensively evaluated in the previous session and was therefore not discussed in detail.

Aspect Explorer was considered as a means to scan quickly for those alternatives which stand out in a positive or a negative way, especially in complicated discussions about multiple viable alternatives *Aspect Explorer* can help to speed up the discussion. Promising or problematic alternatives can quickly be separated. Normally people do not have a similar image in their mind, but Aspect Explorer provides a shared understanding of the discussion topic for all the involved actors. Furthermore it was mentioned that Aspect Explorer should actually be tied to Matchbox. The outcomes of Matchbox are those alternative solutions which are possible or desirable. In Aspect Explorer the same alternatives can be compared on their most important aspects. Therefore the output of Matchbox could be exported as input for Aspect Explorer.

Usability

The ease of use of APS version 2 is a clear improvement over the previous version. However, it remains difficult quickly to find those documents, maps and other information sources that together form the final area plan. Tagging, i.e. labeling with keywords, was mentioned as a possible solution for this.

Although the perceived ease of use of APS2 had improved, it was not considered to be a system in which one can intuitively find one's way. Participants from the infrastructure department indicated that training is would be required to use it. One participant mentioned that "people who use the APS regularly will become skilled in using the APS. You can see the same with respect to RIV [the ports in-house GIS viewer]: some people use it regularly while others do not use it at all."

The participants from the drawing department were more concerned about the ease of use. They mentioned that RIV is a system that is considered to be too complicated by the area planners even though much was invested in increasing its ease of use. In their opinion the APS is RIV plus planning support services. Consequently they doubted whether area planners would be capable of using APS. Their main concern was related to the navigation structure in the portal, not so much to the individual services. They suggested a more wizard oriented user interface that takes the user step by step to an answer.

Regarding the individual services *Map* was considered to be relatively easy to use; *Aspect Explorer* required some explanation but no serious concerns were mentioned. With respect to *Sketchbook* it was mentioned that the concept of virtual folders would require some training.

Usage
Issues that are still relevant and need to be addressed include the authorization of users, who will be responsible for the content etc. Clear policies should be developed for the actual usage of APS within the organization. These policies should, for example, define which information should be annotated in meeting minutes and which information should be annotated in the *Sketchbook*; some information will also need to be archived in a file manager, while other information can be structured in Sketchbook. A lack of clear policies is expected to obscure information availability and thereby reduce the usefulness of APS. Furthermore the availability of services and information should be clearly differentiated, based on user's roles in the planning process.

The participants from the drawing department mentioned training and facilitation as main issues for actual APS usage within the organization. As one participant mentioned: "APS has everything you need, but that is also its greatest weakness". A facilitator from the drawing department could relieve area planners of the burden of having to know APS inside out. Furthermore the participants from the drawing department expect to play an important role in the maintenance of APS because most information is geographic, for example, design sketches of areas in the port will be processed by them. Therefore it seems logical that they also update these designs in APS.

Innovativeness
We especially asked the drawing department for their opinion regarding the innovativeness of APS2, in particular the geographic visualizations.

The digital map is an innovative service especially because of the interactive timeline. Dynamic data and GIS still do not go together easily in GIS applications. Currently there are probably no standard off-the-shelf solutions available. Therefore it was a good idea to base the map on a format such as the one of MapServer. A disadvantage is that MapServer is also still under development; consequently keeping up with their format may be difficult. In contrast to MapServer, which is actually also a producer's format, Open GIS Consortium (OGC) focuses more on feature definitions and projections, however they are also probably considering map formats.

8.5 Conclusions

We can now draw conclusions from the three angles that we used to evaluate APS2: the meeting with the Environmental Department, the interview of the independent PSS expert, and the sessions at the PoR. In general there was a positive correlation between the three angles, which means that we did not find significant contradictions between the different sessions.

The participants of the evaluation sessions at the PoR agreed that APS2 was an improvement over APS1: there was a better balance between the support for process (actions, decisions) and content (design, evaluation), and the ease of use was improved. Overall the participants were convinced of the usefulness of APS, therefore their attention was directed much more towards perceived usefulness and their attitude towards usage.

With respect to usefulness, the phrase “memory of the planning organization” was mentioned several times by different participants. The APS changes the way of thinking about area planning as the actors involved become more aware of the information involved. Clearly the APS is considered as a suite that brings together information from different fields during an area planning process. Even before an area planning project starts, valuable information can already be prepared in APS. For example, safety contours and noise maps can be updated on a regular basis such that crucial information is available and up-to-date at any time. During a project experts can prepare data and make it available through APS during an area planning meeting.

The visualization of temporal and location based information is strongly supported in the suite. APS supports tracing back information, decisions, actions, etc. through the history of the planning project. Furthermore the dynamic *Map* can be used to visualize how an area changes over time, which other projects are planned and other dynamic information.
Furthermore *Sketchbook* was conceived to be a useful means to combine, structure and compare different types of information in a single medium. *Aspect Explorer* was considered to be a useful addition to *Matchbox* to evaluate alternative solutions. *Aspect Explorer* is expected to streamline the debate between experts in different roles during an evaluation of alternative solutions as it facilitates a shared image of the situation.

In general the participants agreed that, compared to APS1, in APS2 the ease of use was improved. As yet APS2 is not a suite through which one can easily navigate without prior training but it should be mentioned that the APS2 walkthrough highlighted all the features in the suite. In a real mainport planning process APS2 services will be offered based on user roles. The participants from the infrastructure department were considerably more positive than the participants from the drawing department. The participants from the infrastructure department, who actually represented the area planners, were of the opinion that, with proper training, APS2 can be usable, the participants from the drawing department doubted this.

This brings us to the attitude towards usage. The Environmental Department and the infrastructure department can be regarded as users of APS, while the drawing department aims to become the maintainer of APS and the data stored in it. Departments involved in area planning will become customers of the drawing department. Consequently the drawing department feels responsible for APS and its intended users. They currently also maintain the RIV system and have seen that it is not used extensively in area planning, which also raises concerns regarding usage of the APS.
Apart from the usability concerns, the participants agreed that policies should be developed to structure the area planning process supported by APS. Further research is required into these policies, which focus on which information should be made accessible, and who may access or modify certain information, etc.

Finally we had the opportunity to ask expert opinions about the innovativeness of our solution, which provided us with evidence that our design decisions also have a contribution to the state of art. The combination of an interactive timeline and geovisualizations was considered to be innovative by both the PSS expert and the experts of the drawing department. Furthermore it was mentioned that the qualitative visualizations offered by

Aspect Explorer add a new angle to planning support, which commonly focuses more on quantitative evaluations. In addition, the introduction of PSS in port planning was considered to be innovative.

9 Epilogue

In the first chapter of this thesis we stated that our objective was to design a suite of services to support studio-based mainport planning. We observed that due to the abundance of information that is used and produced in mainport planning, combined with recent changes in how mainports do business on the global market, inter disciplinary teams of mainport planners require support systems to help them with effective mainport planning. We based this research on the hypothesis that suites of services, applied in studio-based planning processes, can be used as an enabling technology to improve the effectiveness of mainport planning. We stated our main research question as:

Can we create a suite of services to support the actors in a studio-based planning process that improves their effectiveness in mainport planning?

In order to be able to answer this main research question we formulated a number of sub-questions in chapter 2. Answering each of these sub questions brings us a step closer to answering the main question. The big puzzles are built of small pieces. In section 9.1 we discuss and answer our research questions. Next, in section 9.2, we reflect on our application domain and research approach, followed by recommendations for further research in section 9.3.

9.1 Research findings

We discuss our research findings by discussing and answering our research questions in this section. We first answer the sub-questions which were stated in chapter 2, then we answer the main research question.

Research question 1: *What principles in the field of visualization can we use in the design of a suite to support mainport planning?*

We studied literature in the field of visualization to find the answer to this question. In our search for an answer we were guided by the insight and understanding gained during our exploratory case studies. We found that the field of visualization is closely related to the cognitive sciences. However, our objective was not to explain the relation between visualizations and human cognition. We adhered to Card et al. (1999) in that our aim was make use of “computer-supported, interactive, visual representations of data to amplify cognition”.

We found a number of relevant subfields of visualization:

- **scientific visualization**, which is based on the visualization of physical data
- **information visualization**, which is based on the visualization of non-physical data
- **knowledge visualization**, which is focused on the transfer of knowledge between different stakeholders using visualization tools
- **geovisualization**, which addresses the visualization of cartographic information

These subfields are not exclusive but have partly overlapping characteristics. Over the years a variety of classification schemes have been proposed to increase our understanding of the principles underlying visualization. We derived a number of principles that we can follow to design a suite to support mainport planning. We adhered to the reference model for visualization of Card et al. (1999), which provides a high level view on the process of information visualization. On a more detailed level we followed Shneiderman (1996) regarding the classification of data types and interactive tasks which users can execute on visualized data. We adhered to Slocum et al. (2005) who provide a categorization of cartographic data and who provide an overview of the principles of symbolization. Furthermore, because multiple stakeholders typically participate in mainport planning projects, we adhered to the knowledge visualization framework of Burkhard (2004a,b) to address the use of visualizations to transfer knowledge between different stakeholders.

As we observed in chapter 2, mainport planners have to consider dynamic effects and the transition from the current situation towards possible future situations. We concluded that advances in computer graphics can be used to animate dynamic information in mainport planning. Although animation is not yet extensively addressed as a separate topic in information visualization literature, animation principles are extensively described in literature regarding classic dramatic media (Pocock and Rosebuck 2001, Lasseter 1987). We found that animation follows the same lines as the process of visualization presented by Card et al. (1999). Furthermore we found that animation has become an accepted part of computer simulations (O'Keefe 1987, Bishop and Balci 1990).

Visualizations will eventually be embedded in the graphical user interfaces (GUI). This is where the field of visualization meets the field of human computer interaction (HCI). We concluded that we can follow the principles of visualization, but to use these principles effectively in our design we also have to consider the principles of GUI design (Shneiderman 1998).

Research question 2: *What are promising technologies which we can use to implement the identified principles in the field of visualization?*

To answer this question we combined the visualization reference model of Card et al. (1999) with the knowledge visualization framework of Burkhard (2004a,b) in chapter 4. We used the resulting framework, combined with observation of mainport planning in practice (chapter 2), to find promising technologies for our design.

Following the first step in Burkhard's (2004a,b) framework, the visualization objective, we defined our mainport planning suite as an extension of a simulation and modeling suite, which should support the tasks of analysis, design, evaluation and choice in mainport planning projects. Furthermore the suite should support the integration with existing domain specific services.

The second step, following Burkhard (2004a,b), was to identify the type of knowledge which should be exchanged by means of visualization. Following the model of Card et al. (1999), this step starts with accessing raw data. Following today's standard for accessing distributed data sources; we argued that our suite should make use of Uniform Resource Identifiers (URIs). Furthermore, we argued that we should make use of an indexing system to keep track of, and structure, the identified data.

The third step was to, adhering to Card et al. (1999), to structure, filter and transform the identified data. In the literature we found that hierarchical tree structures, which are commonly used in computer's file systems, are suitable for archiving information, but are not optimal for working with large quantities of information on a regular basis. Several alternatives were introduced such as Lifestreams (Fertig et al. 1996b), dynamic visual queries (Shneiderman 1994) and semantic file systems, i.e. using virtual folders (Jones 1999, Cole et al. 2000). We argued that, considering developments in the software industry, a semantic file system is the preferred way of structuring and filtering information. For a semantic file system to function, it should be supplied with semantic information. We argued that an image says a thousand words, but that it can only be correctly interpreted when seeing it in the right context. In mainport planning visualizations are supplemented with text: annotations, explanations and descriptions. We concluded that technologies such as blogging and wikis should be supported in our design to facilitate semantic information.

The fourth step was to map visual structures onto the transformed data (Card et al. 1999), according to a selection for a visualization type (Burkhard 2004a,b). We argued that in mainport planning many visualization types should be supported depending on the context, and the roles of the actors involved. Furthermore, we learned that cartographic visualizations are commonly used as a shared viewpoint for all involved actors (chapter 2). To realize different visualizations in our design, we concluded that we can make use of visualization libraries or existing applications depending on availability and applicability.

The fifth and last step was to create interactive views (Card et al. 1999). Which view is provided to whom depends on the role and tasks of the recipient (Burkhard 2004a,b). Interactive views, in our case, are graphical user interfaces (GUIs) which (re)present visualizations. Key to the workings of modern GUIs is the model-view-controller framework. We argued that our suite should support interactive views for users who work dispersed over a mainport organization. We concluded that we required web-based MVC framework technologies such as AJAX to design rich clients for our suite. Furthermore we concluded that enterprise information portals could be used to provide interactive views depending on a user's role and tasks within a mainport planning project.

Research question 3: *Can we design and implement a suite to support studio-based mainport planning using an architecture which is based on loosely coupled services for visualization and interaction?*

To answer this research question we specified a set of requirements based on the needs of mainport planners (chapter 2) and promising technologies to address these needs (chapter 4). We refer to chapters 5 and 6 for the design of our Mainport Planning Suite (MPS) , and the description of the prototype implementations in chapters 7 and 8. We made a distinction between building blocks and services. In requirement 1 we stated that a MPS should provide services for analysis, design, evaluation, and choice. Each of these services uses several of the technologies which we identified in chapter 4. We argue that the distinction between building blocks and services makes it possible to reuse the same visualization and interaction technologies in different services.

To achieve a loose coupling between the designed services we advocated using the container concept. Containers, such as a portlet container, provide life-cycle management, security, communication, runtime functionalities and other common functionalities that are not specific for individual services. Implementing a container interface ensures interoperability and loose coupling between services and the underlying software platform through clearly defined standards. Hence, we argued that in our MPS a building block in itself is not a service, but a combination of building blocks which implement an interface provided by a container is regarded as a service.

In our MPS design decoupling of concerns is also achieved at other levels of granularity than the services level. For example, we argued that our MPS should be designed as a portlet application (requirement 8) and that a MPS should provide rich clients for user interaction (requirement 7). These two requirements represented a trade-off between using thin server-based MVC frameworks and rich client-based MVC frameworks e.g. using AJAX in a web browser. The choice depends on the type of functionality provided by individual visualization components. We designed the Portlet Event Library to achieve a loose coupling between client-side and server-side MVC frameworks.

Another example of a loose coupling is the link between our MPS and the information services which are available in the mainport organization. We argued that *Sketchbook*, by using virtual folders to show indexed data, established a loose coupling to distributed information sources. Furthermore, we demonstrated that *Map* can link to, and visualize information about port customers which is stored in a database of the commercial department.

By making use of the Java Portlet Standard we also achieved a loose coupling between our suite and the web portal on which the suite is deployed. In chapters 7 and 8 we demonstrated that we could deploy the same mainport planning services on two different web portals. We argue that this decoupling between our suite and the underlying platform makes it easier to integrate a MPS in an existing mainport IT infrastructure.

We conclude that we positively answered our third research question.

Research question 4: *Can we provide evidence that such a suite improves the effectiveness of the actors in studio-based mainport planning?*

To answer this research question we evaluated two prototypes of our MPS at the Port of Rotterdam, respectively named APS1 and APS2. We defined effectiveness in terms of perceived usefulness, perceived usability and attitude towards usage. The outcomes were, as far as applicable, compared to the current way of area planning at the PoR. Based on the outcomes of the APS1 and APS2 evaluation sessions we can answer our research question in the form of “Yes, provided that the following conditions are met.”

We argued that we could provide evidence that our MPS was perceived as a useful and innovative contribution to mainport planning. We argued that situation awareness was an important aspect of usefulness. We concluded, based on the outcomes of the questionnaires, discussions and interviews, that our MPS positively contributed to the awareness of the information used and produced in area planning. An MPS is expected to change the way of thinking about area planning in the port: it will lead to thinner, more to the point plans which can be updated in a “rolling manner”, i.e. at regular intervals. Furthermore the participants of the evaluation sessions said that according to them the MPS is also useful in planning projects other than area planning. Experts in the port clarified the added value of an MPS by referring to it as a “memory of the organization”.

With respect to usability we found that designing an easy to use MPS is a balancing act. On the one hand a MPS should support the structuring and archiving of the information which is used and produced during the planning process, but on the other hand an MPS should also stimulate the creativity needed to find alternative solutions. The participants of the APS1 and APS2 evaluation sessions said that they felt that the bureaucracy introduced by an MPS could potentially harm their creativity. Especially, regarding APS1 they felt that the process support services e.g. action lists and decision lists districted them from the content too much. However, the same participants also recognized the need for a more structured and rational approach towards area planning. We argue that the support offered by an MPS should be carefully tuned to the roles of individual mainport planners, i.e. a secretary should be provided with different services than, for example, a GIS analyst.

We used participatory design as one of our main research instruments. We argued that our iterative design approach, in close cooperation with mainport planners and experts in the PoR, positively contributed to their attitude towards usage of an MPS. The response team at the PoR could provide quick insight into best practices, opened doors to departments that would otherwise remain closed, interest people throughout the organization in studio based mainport planning, and provided feedback on design ideas and prototype implementations.

In addition to the above we can draw a number of conclusions with respect to the development of an MPS. We argued that there are at least three different angles from which the design of an MPS can start. The design of the Airport Business Suite (chapter 2) started from the angle of model integration, i.e. the focus was on integrating a consistent set of existing and commonly used models to compute several performance aspects of an airport. We found that starting from the angle of model integration leads to a need for data integration: to get viable results from an integrated suite of models consistent and accurate data is required. Hence, one could argue that the design of an MPS should start from another angle, namely with the design of a consistent and up to date "project database". In reality, however, we found that the available data falls under the responsibility of different departments and actors, is made for different purposes, and is not always kept up to date. We argued that creating a consistent project database which serves as the input for an integrated set of models, leads to an MPS design in which the focus is no longer on facilitating mainport planners, but merely on maintaining system integrity. We concluded that an MPS design should be focused on supporting mainport planners in visualizing what information they have available even if the available data are outdated and was made for different purposes. In other words, an MPS should be designed with the idea in mind that mainport planners are clever enough to work with information that is not entirely fit to the purpose for which it is used.

An MPS should not be aimed at replacing existing and commonly used design and drawing tools. During the APS1 evaluation session we demonstrated the possibility of digital sketching on geographic maps. Clearly, from the discussion that followed and the interviews with the individual participants, we learned that using pen and paper would remain the preferred way of making sketches. The mainport planners felt that digital sketching would limit them in expressing their creative ideas.

The argumentation above supports our conviction that an MPS, as we designed it, improves the effectiveness of the disciplinary and inter-disciplinary actors in studio based mainport planning when taking into account the identified organizational hurdles. However, when considering the limitations of our MPS evaluation sessions at the Port of Rotterdam, we have to be careful not to generalize the outcomes of this research.

9.2 Research approach

In this section we reflect on our research approach, which was shaped by the choices we made regarding research philosophy, research strategy, and research instruments.

We chose to follow a design science philosophy, i.e. we followed a socio-technologist or developmentalist paradigm. A design science philosophy is indissolubly connected to a design science strategy. Our choice was based on our research objective, the creation of an artifact that should lead to improvement. We took the role of the architect, the designer of the artifact. Being the architect our question was “What is effective?” Due to the synthetic nature of design, the answer to our question could not be found deductively. Our research revolved around an iterative design cycle in which we built and evaluated prototypes of our artifact, the MPS. We relied on mainport planners and domain experts to challenge our design from interpretivist and positivist points of view. In line with the developmentalist paradigm, we used the anomalies that came to light to iteratively improve our design. We concluded that a design science philosophy and strategy helped us to achieve our research objective.

We used a combination of literature research and case studies as the main research instruments. We used literature research to initialize the research, to define our starting point. We also used literature research to sharpen our understanding of visualization principles and relevant state-of-the-art technologies. Two exploratory case studies were used to determine what is needed for effective mainport planning. Clearly two case studies are not adequate for doing a statistical analysis, but we used these case studies to sharpen our insight into and understanding of studio-based mainport planning in practice.
Furthermore we designed and evaluated our MPS in close cooperation with mainport planning experts, using participatory design workshops and prototype demonstrations. We evaluated our MPS using a storyboard and structured walkthrough, questionnaires, group discussion, and half open interviews. Because the results of our MPS evaluation were based on two cases within the same organization, the PoR, the outcomes of our research could be biased. To address this danger, we supplemented our research with an independent expert interview. Furthermore, we made sure that we involved people from several departments of the PoR, all of which had different stakes in the mainport planning process. Yet, we have to be careful when generalizing the results of this research. The results of our research cannot be generalized outside the scope of the context in which they occurred because of the limited number of case studies.

9.3 Future research

We introduce our recommendations for further research.

In chapter 4 we presented a MPS as an extension of simulation and modeling services. During the ABS case study we aimed to integrate simulation models directly into our suite. In the early architecture of the suite, see Figure 14, we also presented simulation and modeling services such as an aspect model adapter, and data adapters. Yet, at the PoR we found that simulation models are currently not used directly in mainport planning meetings. Consequently the focus of our MPS was not on the integration of simulation models. Area planners at the port told us that direct use of simulation models could improve the effectiveness of their planning meetings, but they still lacked the means for this. Further research in this direction would answer the research question:

How can simulation models, such as traffic models, be used directly in studio-based mainport planning meetings?

Our MPS targeted the production of a mainport plan i.e. it supported activities such as analysis, design, evaluation and choice. We learned that there is also a need to query existing mainport plans for specific information, and to present and report on the outcomes of such a query in an automated way. For example, representatives of the PoR would like to access an MPS during customer negotiations to look up areas that fit the requirements of a client quickly. This leads us to the following research question:

To what extent can we improve our MPS to support automatic reporting and presenting of mainport plans, based on specific information needs of decision makers?

From our MPS evaluations we found that actual usage of an MPS in a studio based mainport planning process raises many still unanswered questions. Studio-based mainport planning still lacks a structured and generally accepted approach, which is why we ask the following research question:

Can we design an approach for the usage of an MPS in studio-based mainport planning?

The planners in the PoR indicated that our MPS can also probably be used in projects other than area planning. From reading literature in the field of urban planning we concluded that our suite may be applicable for spatial planning projects in general. Further research is required to find evidence to answer this question:

To what extent is our MPS useful in spatial planning projects in general?

References

Abdelnur, A., and S. Hepper. (2003). *JavaTM Portlet Specifi-cation, version 1.0*. Sun Microsystems. Retrieved March 17, 2005 from www.jcp.org/en/jsr/detail?id=168.

Ackoff, R.L. (1978). *The art of problem solving*. Wiley & Sons.

Acton, G.S. (2004). *Metatheory*. Retrieved December 12, 2006 from http://www.personalityresearch.org/metatheory.html.

Alonso, G., F. Casati, H. Kuno, V. Machiraju (2005). *Web Services: concepts, architectures and applications.* Springer, Berlin.

Baarda, B., M. de Goede, J. Teunissen (2005). *Basisboek kwalitatief onderzoek, handleiding voor het opzetten en uitvoeren van kwalitatief onderzoek*. Stenfert Kroese, The Netherlands.

Ball C., Wm. Swedish. (1981). *Upgraded FAA Airfield Capacity Model*. Federal Aviation Authorities, Washington.

Barreau, D., B.A. Nardi (1995). Finding and reminding, file organization from the desktop. *SIGCHI Bulletin*, Vol. 27, No. 3, pages 39-43.

Bederson, B.B. (2001). PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. Symposium on User Interface Software and Technology, In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages: 71-80.

Berners-Lee, T. (1996). Universal *Resource Identifiers, Axioms of web architecture*. Retrieved December 1, 2006 from http://www.w3.org/DesignIssues/Axioms.html.

Berners-Lee, T. (1998). *Uniform Resource Identifiers (URI): general track - RFC 2396*, Retrieved December 1, 2006 from http://www.ietf.org/rfc/rfc2396.txt.

Berners-Lee, T., D. Connolly (1995). *HyperText Markup Language Specification-2.0 - RFC 1866*, MIT/LCS.

Bertin, J. (1983). *Semiology of graphics*. The University of Wisconsin Press, London, England.

Bishop, J. L. and O. Balci (1990). General purpose visual simulation system: a functional description. In *Proceedings of the 1990 Winter Simulation Conference*, pages 504-512, New Orleans, LA, USA.

Blood, R. (2004). How blogging software reshapes the online community. *Communications of the ACM*, Vol. 47, No. 12, pages 53-55.

Bouvier, D.J., K. Computing (2000). *Getting started with Java3D API.* Sun Microsystems, USA.

Brail, R.K., R.E. Klosterman (2001) *Planning support systems integrating geographic information systems, models, and visualization tools*. ESRI Press, 1st edition.

Briggs, R.O., G.J. de Vreede (2001). *Thinklets, building blocks for concerted collaboration, Version 1.0*. Groupsystems.com, USA.

Briggs, R.O., G.J. de Vreede, J. Nunamaker (2003). Collaboration Engineering with thinklets to persue sustained success with group support systems. *Journal of Management Information Systems*, Vol 19., pages 31-64.

Brin, S., L. Page (1998). *The Anatomy of a Large-Scale Hypertextual Web Search Engine*. Computer Networks, Retrieved December 1, 2006 from http://infolab.stanford.edu/~backrub/google.html.

Bruce, A.R. (2003). Toward an understanding of satisfaction with the process and outcomes of teamwork. *Journal of Management Information Systems*, Vol. 19, No. 4, pages 65-83.

Burbeck, S. (1992). *Applications programming in Smalltalk-80(TM): how to use Model-View-Controller (MVC)*. The UIUC Smalltalk Archive, Retrieved December 1, 2006 from http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html.

Burkhard R.A. (2004a). Towards a framework and a model for knowledge visualization: synergies between information and knowledge visualization. The Netacademy, Switserland, also in *Knowledge and Information Visualization, Searching for Synergies. Lecture notes in Computer Science*, Springer, Germany, 2005.

Burkhard R.A. (2004b). *Learning from architects: the difference between knowledge visualization and information visualization*. The Netacademy, Switzerland.

Burtnyk, N., M. Wein (1975). Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation. *Communications of the ACM*, Vol. 19, No. 10, pages 564-569.

Canas, A.J., R. Calff, G. Hill, M. Carvlho, M. Arguedas, et al. (2005). Concept maps: integrating knowledge and information visualization. In *Knowledge and Information Visualization, Searching for Synergies. Lecture notes in Computer Science*, Springer, Germany.

Card, S.K., JD Mackinlay, B Shneiderman (1999). *Readings in information visualization: using vision to think*. Morgan Kaufman, San Francisco, CA, USA.

Carlsson, C. and E. Turban (2002). DSS: Directions for the next decade. *Decision Support Systems*, Vol. 33, No. 2, pages 105-110.

Carrol, J.M. (2001). *Human-computer interaction in the new millennium*. ACM Press, New York, USA.

Carter, G.M. (1992). M.P. Murray, R.G. Walker, W.E. Walker. *Building organizational decision support systems*. Academic Press, San Diego, CA, USA.

Cayzer, S. (2004). Semantic Blogging: spreading the semantic web meme. In *Proceedings by deepX Ltd*.

Chang, B.W., D. Ungar (1993). *Animation: from cartoons to the user interface*. ACM Press, New York, NY, USA.

Chesbrough, H. and J. Spohrer (2006). A research manifesto for services science. In *Communications of the ACM*, Vol. 49, No. 7, pages 35-40.

Chin R.T.H., A.Verbraeck. (2006a). A suite of decision support services to support port planning. In *Proceedings of Creativity and Innovation in Decision Making and Decision Support 2006*, pages 325-339 London.

Chin R.T.H., J.J. Smits, A. Verbraeck, J.W. Weststrate. (2006b). *Area Planning Studio, a decision enhancement studio to support area planning in the Port of Rotterdam*. Port Research Centre Rotterdam-Delft, Delft, The Netherlands.

Chin, R.T.H., S.P.A. van Houten, A. Verbraeck (2005a). Towards a simulation and visualization portal to support multi-actor decision making in mainports. In *Proceedings of the winter simulation conference*, pages 2500-2505, Orlando, FL, USA.

Chin, R.T.H., S.P.A. van Houten, A. Verbraeck, J. Smits, P. Veenstra, J.W. Weststrate , E. Schalkwijk (2005b). *An area planning studio for the Port of Rotterdam*. International Conference on Port-Maritime Development and Innovation, Rotterdam, 2005.

Chin, R.T.H., S.P.A. Van Houten, A. Verbraeck, J. Smits, P. Veenstra, J.W. Weststrate, E. Schalkwijk. (2005c). *A decision enhancement studio for area planning in the Port of Rotterdam*. Port Research Centre Rotterdam-Delft, Delft.

Churchman C.W. (1971). *The design of inquiring systems: basic concepts of systems and organization*. Basic Books, Inc., Publishers, New York, NY, USA.

Cohen, K.R. (2005). What does the photoblog want? Media, Culture and Society, London, *SAGE Publications*, Vol. 27, No. 6, pages 883-901.

Cole, R. P. Eklund, G. Stumme (2000). CEM, a program for visualizing and discovery in email. In *Proceedings of PKDD 2000, LNAI 1910*, pages 367-374, Springer-Verlag Berlin Heidelberg, Germany.

Coutaz J., J.L. Crowley, S. Dobson, D. Garlan. 2005. Context is key. *Communications of the ACM*, Vol. 48, No. 3, pages 49-53.

Crane, D., E. Pascarello, D. James (2006). *Ajax in Action*. Manning Publications, USA.

Creswell, J.W. (2003). *Research design: qualitative, quantitative and mixed methods approaches*. SAGE Publications, London, UK.

Crinnion, J. (1991). *Evolutionary Systems Development*. Pitman Publishing, London, UK.

Cross, N. (1994). *Engineering design methods: strategies for product design, a Wiley series in product development: planning, designing, engineering*. Chichester, UK, 2nd edition.

Davis, F.D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, Vol. 13, No. 3, pages 319-340.

Endrei, M., J. Ang, S. Chua, P. Comte, P. Krogdahl, M. Luo, T. Newling (2004). *Patterns: Service Oriented Architecture and Web Services*. Redbooks, IBM, USA.

Endsley, M.R. (1995). Toward a theory of situation awareness. *Human Factors*, Vol. 37, No. 1, pages 32-64.

Farley, J. W. Crawford, P. Malani, J. Gehtland (2006). *Java Enterpise in a nutshell*. O'Reily, Sabastopol, CA, USA.

Fertig, S., E. Freeman, D. Gelernter (1996a). “Finding and reminding” reconsidered. *SIGCHI Bulletin*, Vol. 28, No. 1, pages 66-69.

Fertig, S., E. Freeman, D. Gelernter (1996b). Lifestreams: an alternative to the desktop metaphor. *CHI96*, pages 410-411.

Foley, J.D., A. Van Dam (1982). *Fundamentals of interactive computer graphics*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Fowler, M., K. Scott (2000). *UML distilled: a brief guide to the standard Object Oriented modeling language*. Addison-Wesley, USA.

Gamma, E., R. Helm, R. Johnson, J. Vlissides (1995). *Design Patterns: elements of reusable Object-Oriented software*. Addison-Wesley, Boston, MA, USA.

Garrett, J.J. (2005). *Ajax: A new approach to web applications*. Adaptive Path, San Francisco, CA, USA, Retrieved December 1, 2006 from http://www.adaptivepath.com/publications/essays/archives/000385.php.

Geertman, S. (2002). Inventory of Planning Support Systems in planning practice: conclusions and reflections. In *Proceedings of the 5th AGILE Conference on Geographic Information Science*, pages 1-5, Palma, Spain.

Gelernter, D. (2003). *Future of operating systems: simplicity*. Computerworld, http://www.computerworld.com/printthis/2003/0,4814,77335,00.html, accessed 1/27/2006.

Genender, J., B. Snyder, S. Li (2006). *Professional Apache Geronimo*. Wrox, Las Vegas, USA.

Gifford, D.K., P. Jouvelot, M.A. Sheldon, J.W.O'Toole (1991). Semantic file systems. *ACM SIGOPS Operating Systems Review*, Vol. 25, Issue 5, pages: 16 - 25.

Goldberg, A., D. Robson (1983). *Smalltalk-80, the language and its implementation*. Addison-Wesley Publishing Company, Boston, MA, USA.

Gout, M.W., R.C.G. Haffner, J. van Sinderen (1997). *Mainports in the 21st century*. Wolters-Noordhoff, Groningen, Nederland.

Gregg, D.G., U.R. Kulkarnu, A.S. Vinze (2001). Understanding the philosophical underpinnings of software engineering research in information systems. In *Information Systems Frontiers*, vol. 2, no. 3, pages 169-183, Kluwer Academic Publishers, The Netherlands.

Hengst, M. den, N. Lang (2002). *Airport Business Suite project: user analysis*. TU Delft, Delft, The Netherlands, report.

Henriksen, J.O., N.J. Earle (1992). Proof animation: the general purpose animator. in *Proceedings of the 1992 Winter Simulation Conference*, pages 176-181, Phoenix, Arizona, USA.

Hepper, S. (2006). *Java[tm] Portlet Specification, version 2.0 early draft - JSR 286*. IBM Corp, USA.

Hevner, A.R., S.T. March, J. Park, S. Ram (2004). Design science in Information Systems research. Accepted for publication in *MIS Quarterly 2004*.

Jacobs, P.H.M. (2005). *The DSOL simulation suite, enabling multi-formalism modeling in a distributed context*. Delft University of Technology, Delft, The Netherlands, Doctoral dissertation.

Jaeschke G., M. Leissler, M. Hemmje (2005). Modeling interactive, 3-dimensional information visualizations supporting information seeking behaviors. In *Knowledge and Information Visualization, Searching for Synergies. Lecture notes in Computer Science*, Springer, Germany.

Jamieson, S (2004). Likert scales: how to (ab)use them. *Medical Education*, Vol. 38, pages 1212–1218, Blackwell Publishing Ltd.

Jonassen, D.H. (2005). Tools for representing problems and the knowledge required to solve them. In *Knowledge and Information Visualization, Searching for Synergies. Lecture notes in Computer Science*, Springer, Germany.

Jones, K. (1999). *View Mail Users Manual*. http://www.wonderworks.com/vm. 1999

Keen P.G.W. , H.G. Sol (2007). *Decision Enhancement Services*. Forthcoming.

Keen, P. G. W., M. S. Scott Morton (1978). *Decision Support Systems, an organizational perspective*. Addison-Wesley Publishing Company, Inc., Philippines.

Keller, A.I., A. Hoeben, A. van der Helm (2006). Cabinet: merging designers' digital and physical collections of visual materials. In *Personal and Ubiquitous Computing*, Vol. 10, No. 2-3, pages 183-186, Springer, London, UK.

Keller, A.I., P.J. Stappers, S. Vroegindeweij (2004). Supporting informal collections of digital images: organizing, browsing and sharing. In *Proceedings of the conference on Dutch directions in HCI*, Vol. 65, Amsterdam, The Netherlands.

Keller, T., S. Tergan (2005). Visualizing knowledge and information: an introduction. In *Knowledge and Information Visualization, Searching for Synergies. Lecture notes in Computer Science*, Springer, Germany.

Knudsen, J.B., Knudsen J. (1999). *Java2D Graphics*, O'Reilly Media, USA.

Krasner, G.E., S.T. Pope (1988). A cookbook for Using the model-view-controller User Interface paradigm in Smalltalk-80. *Journal of Object-Oriented Programming*, Vol. 1, No. 3, pages 26-49.

Kuipers, B. (1999). *Flexibiliteit in de Rotterdamse havenregio*. University of Groningen, Groningen, The Netherlands, Doctoral dissertation.

Larman, C. (2004). *Applying UML and Patterns, an Introduction to Object-Oriented analysis and design and iterative development*. Prentice Hall, Upper Saddle River, NJ, USA.

Lasseter, J. (1987). Principles of traditional animation applied to 3D computer animation. In Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pages 35-44.

Law, A., W. Kelton (2000). *Simulation modeling and analysis*. Mc Graw Hill, Singapore, Singapore, 3rd edition.

Leuf, B., W. Cunningham (2001). *The Wiki way: quick collaboration on the Web*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Lugt, R. van der (2001). *Sketching in design idea generation meetings*. Delft University of Technology, Delft, The Netherlands, doctoral dissertation.

Mackinlay, J.D., G.G. Robertson, S.K. Card (1991). The perspective wall: detail and context smoothly integrated. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, pages 173 - 176.

Malczewski, J. (1999). *GIS and multicriteria decision analysis*. John Wiley & Sons, Inc. USA.

Mann, H. B. and Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*. Vol.18, pages 60-60.

March, S.T., G.F. Smith (1995). Design and natural science research on information technology. *Decision Support Systems*, vol. 15, pages 251-266, Elsevier Science B.V., The Netherlands.

Margolis, E. and S. Laurence (2003). *Concepts: core readings*. MIT Press, USA.

McCormick B.H., T.A. DeFanti, M.D. Brown (1987). *Visualization in scientific computing—a synopsis*. ACM SIGBIO Newsletter, Vol. 10, No. 1, pages 15-21.

Menzel,C., R.J. Mayer (1998). *The IDEF family of languages*. Handbook on Architectures of Information Systems, philebus.tamu.edu. Retrieved January 22, 2007 from http://philebus.tamu.edu/~cmenzel/Papers/idef-family.pdf.

Miser, H.J. (1985). E. Quade. *Handbook of systems analysis; overview of uses, procedures, applications, and practice*. Wiley, Chichester, UK.

Mitroff, I., F. Betz, L. Pondy, F. Sagasti (1974). On managing science in the systems age: two schemas for the study of science as a whole systems phenomenon. The Institute of Management Sciences, *Interfaces*, vol. 4, no. 3, pages 46-58.

Mongkolwat, P., A. Kogan, J. Koh, D.S. Channin (2005). Blogging your PACS. *Journal of Digital Imaging*, Vol. 18, No. 4, pages 326-332.

Neufville, R de, A. Odoni (2003). *Airport systems: planning, design, and management*. McGraw-Hill Professional, USA.

Nielsen, J. (1999). User interface directions for the Web. *Communications of the ACM*, Vol. 42, No. 1, pages 65-72.

Nottingham, M., R. Sayre (2005). *The Atom syndication format - RFC4287*. Retrieved December 1, 2006 from http://www.ietf.org/rfc/rfc4287.

Novak, J., M. Wurst (2005). Collaborative knowledge visualization for cross-community learning. In *Knowledge and Information Visualization, Searching for Synergies. Lecture notes in Computer Science*, Springer, Germany.

O'Keefe, R. M. (1987). What is Visual Interactive Simulation? (And is there a methodology for doing it right?). In *Proceedings of the 19th conference on Winter simulation*, pages 461 - 464, Atlanta, Georgia, USA.

Oosterhuis, K. (2003). *Hyperbodies: towards An E-motive Architecture*. Birkhäuser – Publishers for Architecture, Basel, Switzerland.

Pallant, J. (2005). *SPSS survival manual*. Open University Press, Berkshire, UK.

Papazoglou, M., J.-J. Dubray (2004). *A survey of web service technologies*. Dit-04-058, Informatica e Telecommunicazioni, University of Trento, Trento, Italy.

Papazoglou M., D. Georgakopoulos (2003). Service oriented computing. *Communications of the ACM*, Vol. 46, No. 10, pages 25-28.

Peterson, M. P. (1995). *Interactive and animated cartography*. Prentice Hall, New Jersey, USA.

Pocock, L., J Rosebush (2001). *The computer animator's technical handbook*. Morgan Kaufman, USA.

Reiners, D. (2002). *Scene Graph Rendering*. OpenSG Forum. Retrieved 2005 from http://vrjuggler.org/pub/scenegraph-rendering.ieeevr2002.pdf.

Robertson G., J.D. Mackinlay, S.K. Card (1991). Cone Trees: animated 3D visualizations of hierarchical information. *In Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology,* pages 189 - 194.

Roozenburg, N. J. M., J. Eekels (1991). *Product design: fundamentals and methods*. John Wiley & Sons, USA.

Schalkwijk, E. (2005). *A process design and tool-suite to enhance matching in the Port of Rotterdam*. Port Research Centre, Rotterdam-Delft, Delft, The Netherlands, Master thesis.

Schiano, D.J., B.A. Nardi (2004). Blogging by the rest of us. In CHI '04 extended abstracts on Human factors in computing systems, pages 1143 – 1146, Vienna, Austria.

Schulze, T., G. Horton, B. Preim, S. Schlechtweg (2005). *Simulation und Visualisierung 2005*. Institut für Simulation und Graphik der Otto-von-Guericke-Universität Magdeburg. Magdeburg, Germany.

Seshadri,G. (1999). *Understanding JavaServer Pages Model 2 architecture, exploring the MVC design pattern*. JavaWorld.com, Retrieved January 19, 2007 from http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc_p.html.

Shan, Y. (1989). An event-driven model-view-controller framework for Smalltalk. In *OOPSLA '89 Proceedings*, pages 347-352.

Shannon, R.E. (1975). *Systems simulation: the art and science*. Prentice-Hall, Indianapolis, IN, USA.

Shneiderman, B. (1994). Dynamic queries for visual information seeking. IEEE Software, Vol. 11, No. 6, pages 70-77.

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336-343.

Shneiderman, B. (1998). *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Simon, H.A. (1955). A behavioral model of rational choice. In *The Quarterly Journal of Economics*, vol. LXIX.

Simon, H.A. (1976). *Four essays on procedural rationality in economics*. Carnegie-Mellon University, Pittsburgh, Pennsylvania, USA.

Simon, H.A. (1977). *Models of discovery and other topics in the methods of science*. D. Reidel Publishing Company, Dordrecht, The Netherlands.

Simon, H.A. (1983). Search and reasoning in problem solving. North-Holland, *Artificial Intelligence*, vol. 21, pages 7-29.

Simon, H.A. (1986). Decision Making and Problem Solving. In *Research Briefings 1986: Report of Research Briefing Panel on Decision Making and Problem Solving*, National Academy Press, Washington, DC, USA.

Slocom, T.A., R.B. McMaster, F.C. Kessler, H.H. Howard (2005). *Thematic cartography and geographic visualization*. Prentence Hall. Upper Saddle River, USA.

Smits, J., P. Veenstra, J.W. Weststrate , R.T.H. Chin, S.P.A. Van Houten, A. Verbraeck, E. Schalkwijk (2005). *A challenge in port planning and design, Towards an Area Planning*

Studio for the Port of Rotterdam. International Conference on Port-Maritime Development and Innovation, Rotterdam, The Netherlands.

Sol, H.G. (1982). *Simulation in information systems development*. University of Groningen, Groningen, The Netherlands, Doctoral dissertation.

Spence, R. (2001). *Information Visualization*. ACM Press, Essex, UK.

Steenbergen, C., H. Mihl, W. Reh, F. Aerts (1999). *Architectural design and composition*. TU Delft, Faculty of Architecture, THOTH Publishers, Bussum, The Netherlands.

Stillwell, J., S. Geertman, S. Openshaw (1999). *Geographical information and planning*. Springer, Heidelberg, Germany.

Strauss, P.S., R. Carey (1992). An Object-Oriented 3D Graphics Toolkit. *Computer Graphics*, Vol. 26, No. 2, pages 341 - 349.

Streitz, N. P. Nixon (2005). The disappearing computer. *Communications of the ACM*, Vol. 48, No. 3, pages 32-35.

Strothotte, C. and T. Strothotte (1997). *Seeing between the pixels, Pictures in interactive systems*. Springer, Heidelberg, Germany.

Stumpe, J. (1997). *Hoeveel ruimte geeft aan luchtvaart, integrale beleidsvisie over de toekomst van luchtvaart in Nederland*. Bv Kunstdrukkerij Mercurius-Wormerveer, The Netherlands.

Sturman D. (1998). The sate of computer animation. ACM SIGGRAPH Computer Graphics, Vol. 32, No. 1, pages 57 - 61.

Tufte, E.R. (1990). *Envisioning information*. Graphics Press, Cheshire.

Vaishnavi, V. W. Kuechler (2006). Design research in information systems. at http://www.aisworld.org/Reseachdesign/drisISworld.htm, last accessed: 10-03-2006.

Visser, H.G., R.T.H. Chin , R.A.A. Wijnen, W.E. Walker, J. Keur, U. Kohse, J. Veldhuis, A.R.C. De Haan (2003). The Airport Business Suite: a Decision Support System for Airport Strategic Exploration. *AIAA-2003-6742, AIAA's 3rd Annual Aviation Technology, Integration, and Operations (ATIO) Technical Forum*, Denver, Colorado, USA.

Vonk, G., S. Geertman, P. Schot (2005). Bottlenecks blocking widespread usage of planning support systems. *Environment and Planning*, Vol. 37, pages 909-924.

Vrede, H.de, E. Nanninga-Kryzewski, J. Knapen (1993). *IMER, Integraal milieu-effect rapport Schiphol en omgeving*. Van de Rhee, Rotterdam, The Netherlands.

Vujosevic, R. (1990). Object Oriented Visual Interactive Simulation. In *Proceedings of the 22nd conference on Winter simulation*, pages 490 – 498, New Orleans, Louisiana, USA.

Walker, W.E. (2000). Policy Analysis: A Systematic Approach to Supporting Policymaking in the Public Sector. *Journal of Multicriteria Decision Analysis*, Vol.9, No. 1-3, pages 11-27.

Walker, W.E, N.A. Lang, J. Keur, H.G. Visser, R.A.A. Wijnen, U. Kohse, J.Veldhuis, A.R.C. De Haan (2003). An Organizational Decision Support System for Airport Strategic Exploration, In *Tung Bui, Henryk Sroka, Stanislaw Stanek, and Jerzy Goluchowski (eds.), DSS in the Uncertainty of the Internet Age*, pages 435-452, Publisher of the Karol Adamiecki University of Economics in Katowice, Katowice, Poland.

Ware, C. (2005). Visual Queries: the foundation of visual thinking. In *Knowledge and Information Visualization, Searching for Synergies. Lecture notes in Computer Science*, Springer, Germany

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics*, Vol. 1, pages 80-3.

Yin, R.K. (1994). *Case study research: design and methods*. Sage Publications, Thousand Oaks, CA, USA, 3rd edition.

Ziebold, G., M. Sum (2006). Best *Practices for Applying AJAX to JSR 168 Portlets*. Sun Developer Network, Retrieved January 22, 2007 from http://developers.sun.com/prodtech/portalserver/reference/techart/ajax-portlets.html.

Zeigler, B. (1976). *Theory of modeling and simulation*. Academic Press, San Diego, CA, US

Appendix A: Libraries

A1: Virtual folder library

An UML class diagram of the virtual folder library is shown in Figure 84. A virtual folder can be of type BOX or of type FILTER. As explained in section 6.3.3 a BOX-type is a virtual folder which is used to group content on virtual stacks, while a FILTER-type is used to automatically filter content based on meta-data. When a virtual folder is a BOX-type then it has zero *WhereStatement*s. When a virtual folder is a FILTER-type then it has zero *fileId*s. Our implementation leaves room for a combined BOX-FILTER-type, however we believe that such a type would confuse a user of the exact function of a virtual folder.

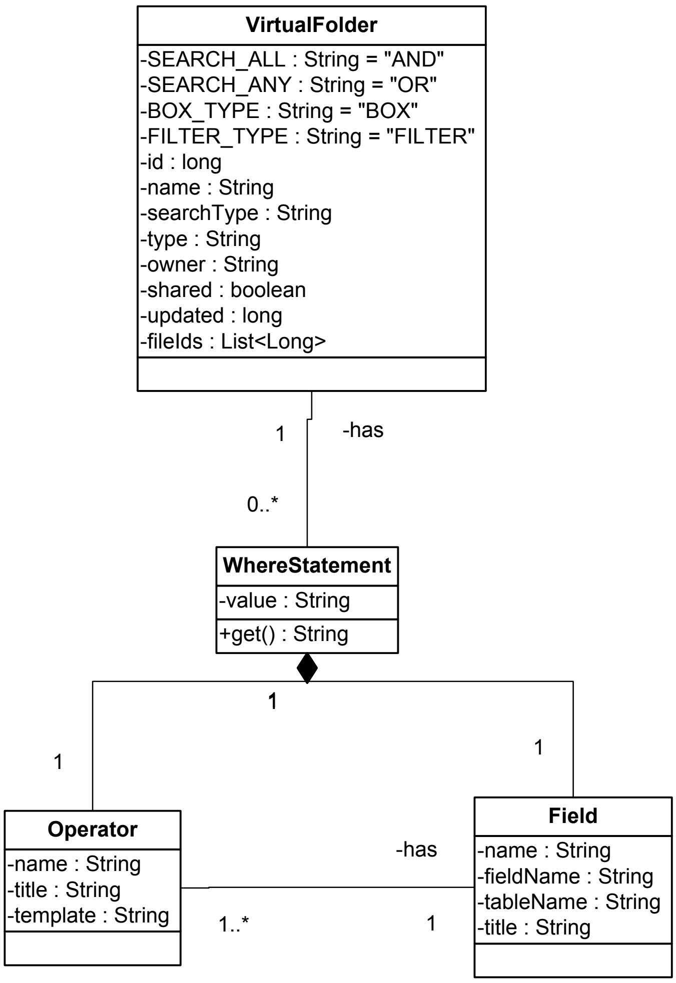


Figure 84: An UML class diagram of the virtual folder library.

A *WhereStatement* represents a where statement as used in an SQL query. For example:

SELECT * FROM file WHERE file.name LIKE ‘test’

Multiple *WhereStatement*s can be specified and combined with AND, or OR operators, i.e. to search according to all criteria or to any criteria. For example:

SELECT * FROM file WHERE file.name LIKE ‘test’ **AND** file.size < 5000

A *Field* represents a field in a database table, e.g. file.name. An *Operator* represents an operator and a template such as LIKE '{value}', where {value} is replaced by the actual value specified in the *WhereStatement* e.g. "test". A *Field* also has a list of available *Operators*, which is used in the GUI to enable a user to select an *Operator* which he or she wants to use. The selected operator is then used by the *WhereStatement*. Note that a *Field* contains a *tableName* as an attribute while a *WhereStatement* does not. This enables us to specify more complicated queries which combine the results of multiple database tables. Furthermore in *Field*s and *Operator*s *name*-attributes are used as handles for processing and *title*-attributes are used as a label in the GUI.

The available filters which can be used in virtual folders are specified in an XML-file, for example:

```
<virtualFolderFilters>
      <field>
            <name>file.name</name>
            <fieldName>name</fieldName>
            <tableName>file</tableName>
            <title>file name</title>
            <operator>
                  <name>is</name>
                  <title>is</title>
                  <template>LIKE '{value}'</template>
            </operator>
            <operator>
                  <name>isNot</name>
                  <title>is not</title>
                  <template>NOT LIKE '{value}'</template>
            </operator>
            <operator>
                  <name>startsWith</name>
                  <title>starts with</title>
                  <template>LIKE '{value}%'</template>
            </operator>
            <operator>
                  <name>endsWith</name>
                  <title>ends with</title>
                  <template>LIKE '%{value}'</template>
            </operator>
            <operator>
                  <name>contains</name>
                  <title>contains</title>
                  <template>LIKE '%{value}%'</template>
            </operator>
      </field>
      <field>
      ...
</field>
</virtualFolderFilters>
```

The GUI used for specifying a virtual folder is shown in Figure 85. In this example a virtual folder is created to display all the geographic maps that were made between 01-10-2006 and 01-11-2006.

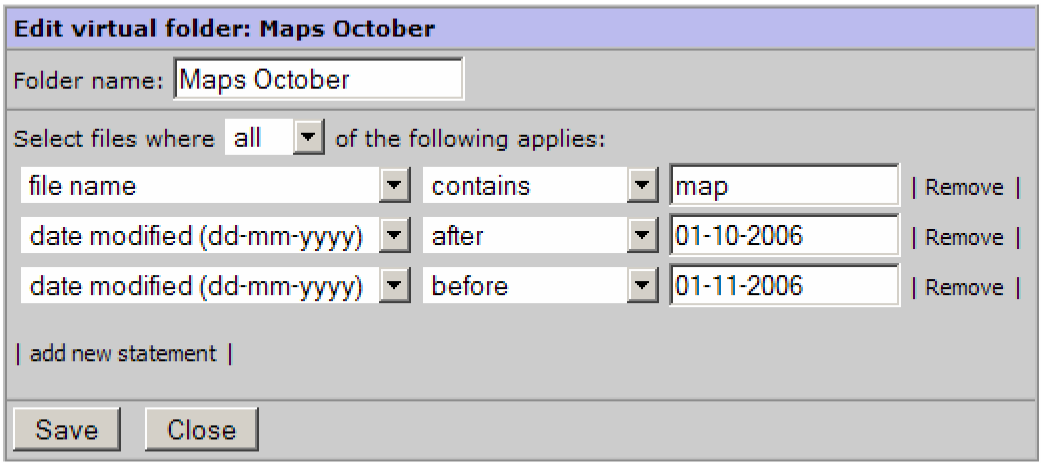


Figure 85: Specification of a virtual folder in a GUI.

A2: Indexer library

An UML class diagram of the indexer library is shown in Figure 86. The role of the indexer library is to create an index of remote libraries such as file system folders. A *Library* represents such a remote library which is referenced through a URI (see section 4.2). An *IndexedFile* is a file stored in a library. The indexing itself happens by *TargetIndexer*, which has two functions: one, to create the index in a (mySQL) database, and two, to render thumbnail images of each file. The thumbnail images are generated by a renderer which implements the *RendererInterface* depending on the MIME-type of the file.

Because the indexing of large libraries can take a considerable amount of time, we specified an *IndexerThread*. The *IndexerThread* instantiates a singleton of itself, and it creates a queue of libraries and individual files which should be indexed, and it processes these one by one to create the index. We made a distinction between complete libraries and individual files, because when a library was already indexed only changes to individual files should be updated in the index. Individual files waiting to be indexed are specified in a *FileItem*, which stores the URI of the file and the library to which it belongs. *IndexerInfo* is an object which provides indexing progress information to a user.

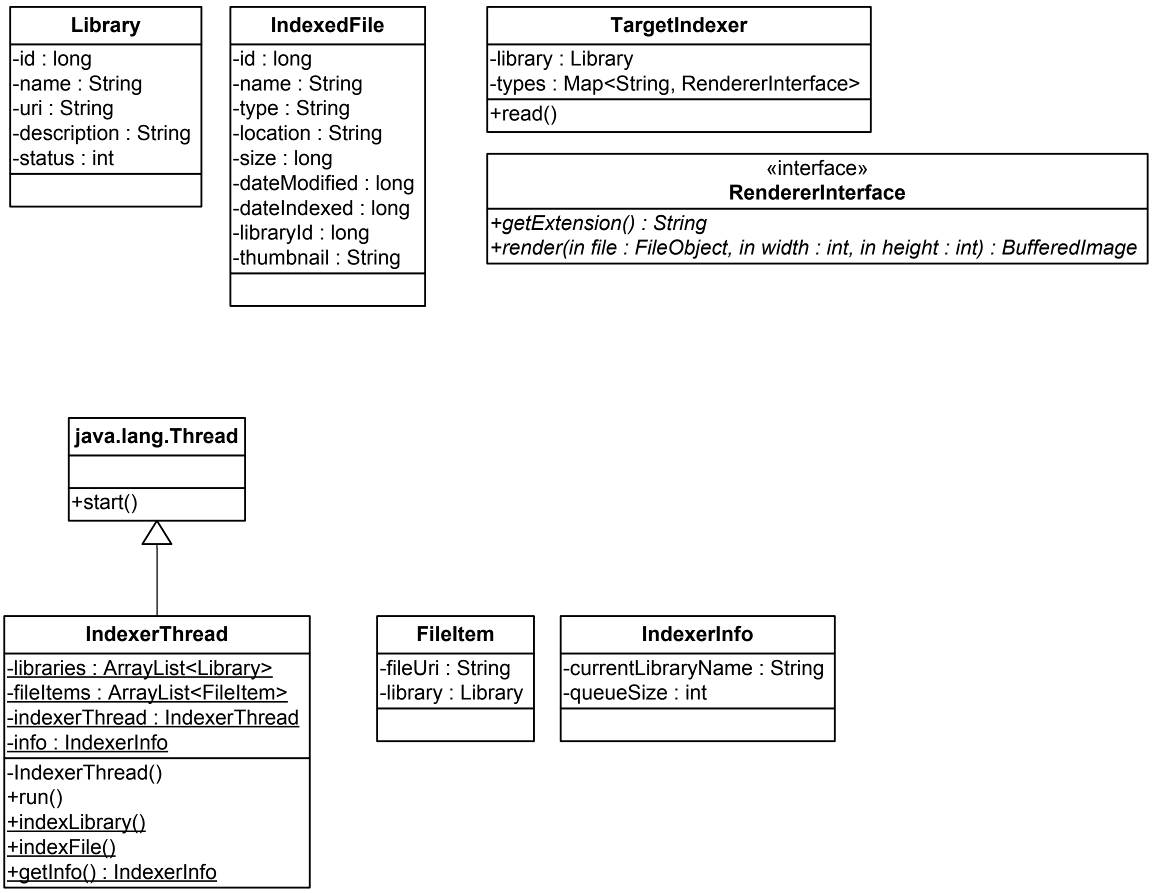


Figure 86: A UML class diagram of the indexer library.

The workings of the *IndexerThread* are further explained by the UML sequence diagram shown in Figure 87. The diagram shows the indexing sequence for a library. First the *IndexerThread* instantiates a singleton of itself, the *indexerThread*. The *indexerThread* starts up the thread, which keeps looping through queued libraries and files to index. Suppose a user specifies a new library in a portlet, the *indexerPortlet*. Next this library is added to the end of the queue of libraries waiting to be indexed. When the specified library is the first in the queue, a *targetIndexer* is called to read the library and create an index of it. Next the *info* object is updated to provide user feedback about the progress of the indexing process. Finally the library is removed from the queue and the *indexerThread* continues with the next library in the queue if any.

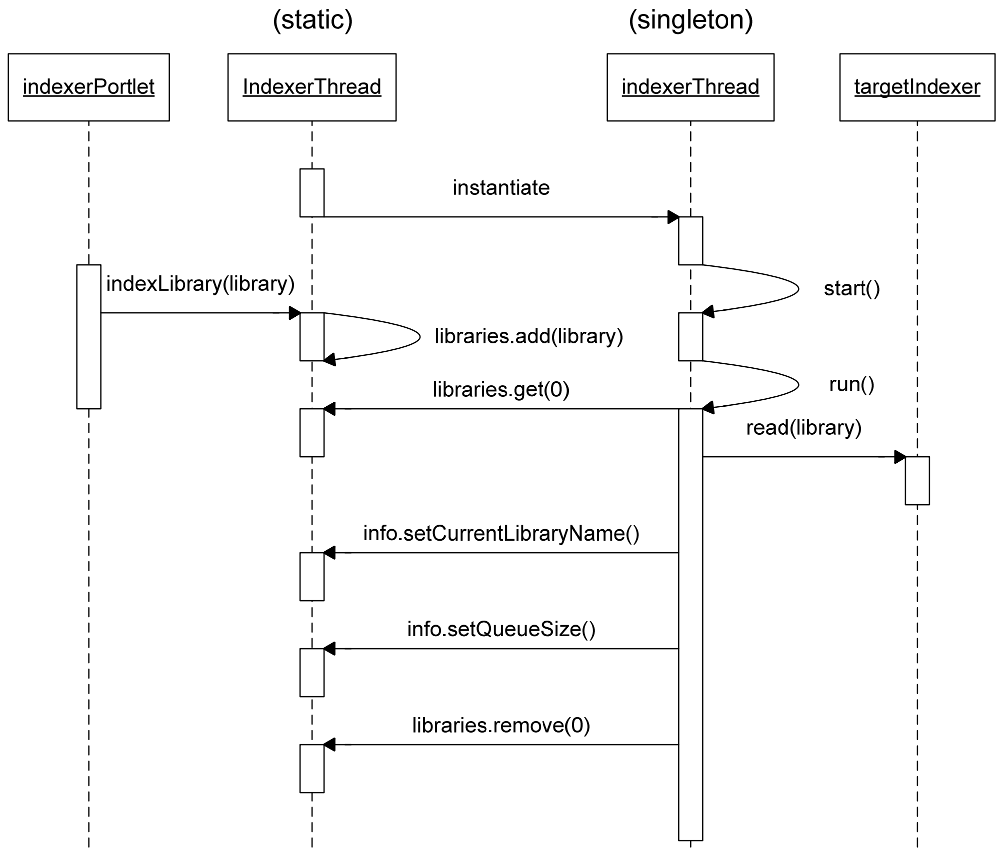


Figure 87: An UML sequence diagram of the indexing thread.

An example of an indexerPortlet is shown in Figure 88. Each library has an unique name, an URI, a description and a status. The URIs shown in the example refer to a local file system for testing purposes. The first library has the default status, which means that new files which are added will by default end up in that library. Libraries can also be disabled to temporarily hide information.

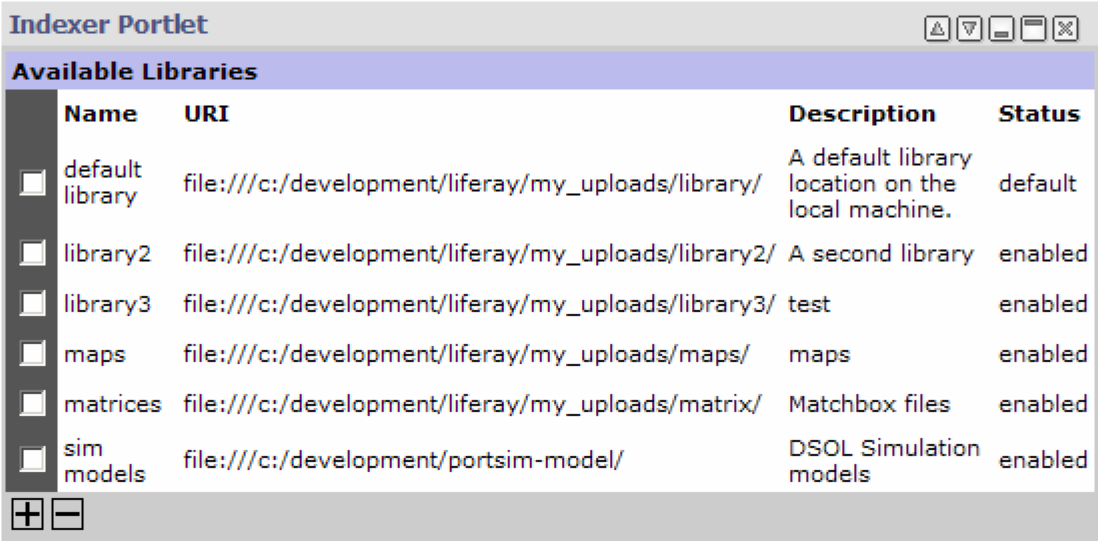


Figure 88: An Indexer Portlet.

A3: Blog library

An UML class diagram of the blog library is shown in Figure 89. The blog library is adds input-output functionality to the off-the-shelf ROME library. ROME is a library which provides objects for creating blog feeds in ATOM and RSS formats. Our blog library enables us to write feeds to a database, read feeds from a database, and erase feeds from a database.

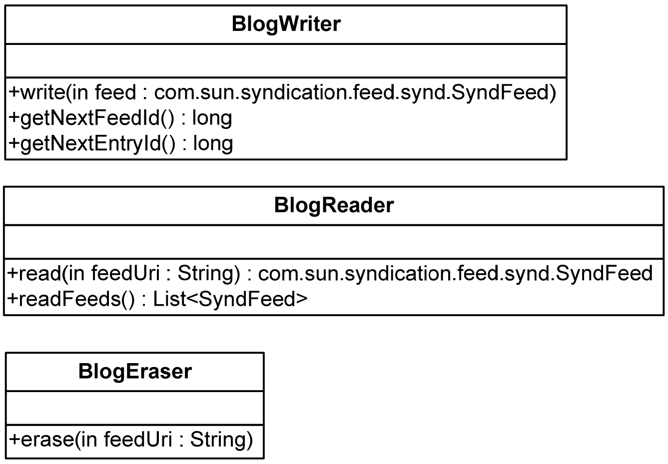


Figure 89: An UML class diagram of the blog library.

An example of the usage of the blog library in a portlet is shown in Figure 90. This example shows the annotation of a map. The “Link” opens the map in the *Map* service, and the “Atom feed”-link enables a user to add the feed to feed-readers.

Figure 90: An example of using the blog library in a portlet.

A4: GUI library

The GUI library provides a number of GUI components which were specifically developed for our MPS. It contains GUI components which are implemented in Java Swing and JavaScript. Components implemented in Java Swing can be used in desktop applications, Java Web Start applications and applets. The JavaScript components can be used in web-browser based user interfaces such as portlets and traditional webpages.

Time slider is an example of a GUI component which was implemented both in Java Swing and in JavaScript see respectively Figure 91 and Figure 92. The time slider is a GUI component used to specify a time interval by dragging the slider buttons left and right. In the Java Swing example we added controls for selecting a start and end year. In the JavaScript example we did not add similar controls, but the slider itself provides exactly the same functionality.

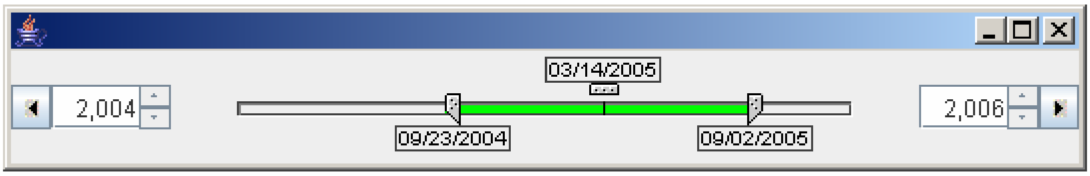


Figure 91: A time slider implemented in Java Swing.

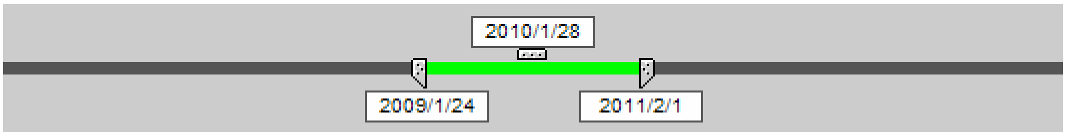


Figure 92: A time slider implemented in JavaScript.

Other JavaScript-based GUI components, which were used in *Sketchbook*, are shown in Figure 93 and Figure 94. In Figure 93 we combined a slider and an image-slider to be able to browse through a set of images. The image-slider shows the available images, or thumbnails, and enables a user to open an image by clicking on it. The slider is used the scroll the image slider left and right. In Figure 94 we combined a time-slider and a map viewer. The map viewer displays a map which is rendered on a server and it enables a user to pan, zoom and click markers. The markers can be linked to a time interval. When this time interval lies within the time interval specified in the time slider then the marker is visible, otherwise it is hidden.

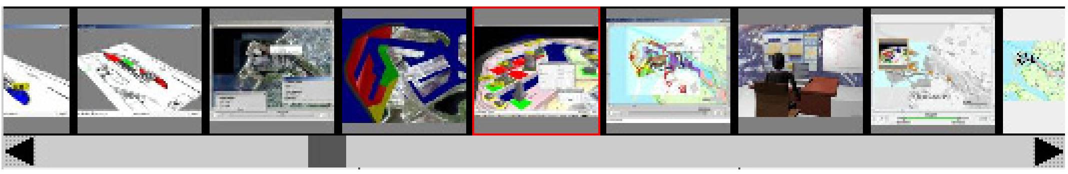


Figure 93: A slider and an image-slider implemented in JavaScript.

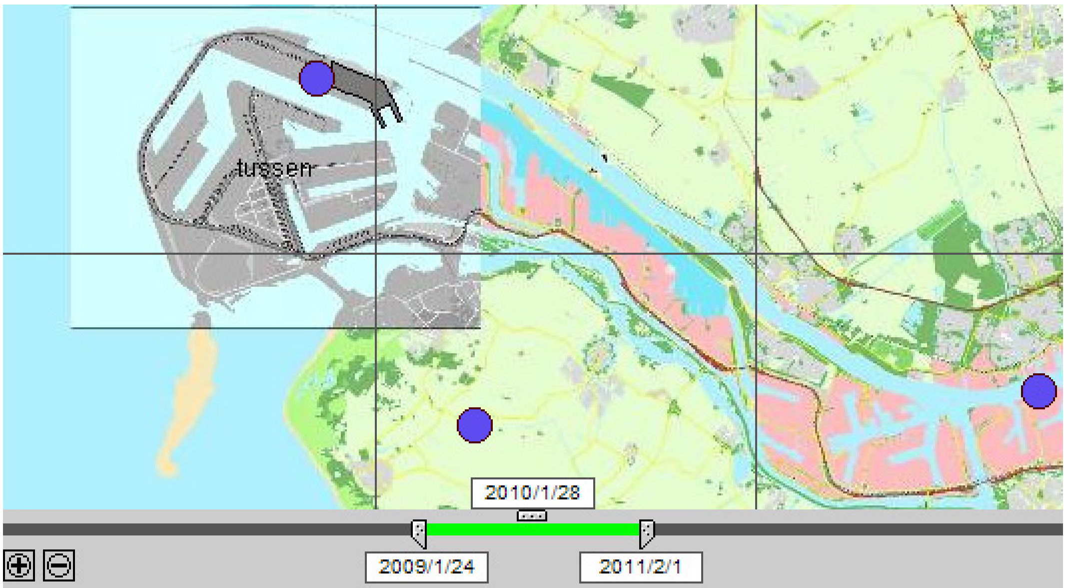


Figure 94: A time-slider and map viewer implemented in JavaScript.

How to create Java Swing components is extensively documented by Sun Microsystems[26]. Using JavaScript to create GUI components is less obvious, hence we now focus on the JavaScript implementations. We used JavaScript to implement web-based GUI components, which are represented in the browser as HTML fragments. The HTML DIV-element is the key to understanding how JavaScript GUI components work. When a webpage contains a DIV-element, then JavaScript code can be used to "change" this DIV-element into a GUI component. A DIV-element defines a rectangular area in a webpage which can be referenced and modified by JavaScript code. We demonstrate this using the JavaScript code of time slider listed below.

```
// Namespace
if (studio == undefined) var studio = {};

/**
 * Class TimeSlider
 * a time slider
 * depends on Prototype
 */
studio.TimeSlider = Class.create();
studio.TimeSlider.prototype = {
  /**
   * Constructor.
   */
  initialize: function(timeSliderId, totalInterval, currentInterval) {
        // Set parameters
        this.timeSliderDiv = $(timeSliderId);
        this.totalInterval = totalInterval;
        this.currentInterval = currentInterval;

        // Size
        //this.left = this._toNumber(this.timeSliderDiv.style.left);
        //this.top = this._toNumber(this.timeSliderDiv.style.top);
        this.width = this._toNumber(this.timeSliderDiv.style.width);
        this.height = 20; // Fixed height
```

[26] See http://java.sun.com/docs/books/tutorial/uiswing/

```
        this.timeSliderDiv.style.height = this.height + "px";

        // Value formatter (by default we format millis. as date)
        this.formatter = new studio.DateFormatter();

        // Change listeners
        this.changeListeners = new Array();
        this.notifyOnDrag == false;

        // Construct the slider
        var sliderDiv = document.createElement("div");
        sliderDiv.style.position = "absolute";
        sliderDiv.style.overflow = "hidden";
        sliderDiv.style.left = "0px";
        sliderDiv.style.top = "7px";
        sliderDiv.style.width = this.width + "px";
        sliderDiv.style.height = "6px";
        sliderDiv.style.backgroundColor = "#555555";
        sliderDiv.style.backgroundRepeat = "no-repeat";
        this.timeSliderDiv.appendChild(sliderDiv);

        this.intervalDiv = document.createElement("div");
        this.intervalDiv.style.position = "absolute";
        this.intervalDiv.style.overflow = "hidden";
        this.intervalDiv.style.top = "7px";
        this.intervalDiv.style.height = "6px";
        this.intervalDiv.style.backgroundColor = "#00ff00";
        this.intervalDiv.style.backgroundRepeat = "no-repeat";
        this.timeSliderDiv.appendChild(this.intervalDiv);

        this.currentButtonDiv = document.createElement("div");
        this.currentButtonDiv.style.position = "absolute";
        this.currentButtonDiv.style.left = "0px";
        this.currentButtonDiv.style.top = "0px";
        this.currentButtonDiv.style.width = "15px";
        this.currentButtonDiv.style.height = "6px";
        this.currentButtonDiv.style.backgroundImage =
    "url(/sketchbook/images/topslider.gif)";
        this.currentButtonDiv.style.backgroundRepeat = "no-repeat";
        this.timeSliderDiv.appendChild(this.currentButtonDiv);

        this.startButtonDiv = document.createElement("div");
        this.startButtonDiv.style.position = "absolute";
        this.startButtonDiv.style.top = "5px";
        this.startButtonDiv.style.width = "8px";
        this.startButtonDiv.style.height = "16px";
        this.startButtonDiv.style.backgroundImage =
    "url(/sketchbook/images/leftslider.gif)";
        this.startButtonDiv.style.backgroundRepeat = "no-repeat";
        this.timeSliderDiv.appendChild(this.startButtonDiv);

        this.endButtonDiv = document.createElement("div");
        this.endButtonDiv.style.position = "absolute";
        this.endButtonDiv.style.top = "5px";
        this.endButtonDiv.style.width = "8px";
        this.endButtonDiv.style.height = "16px";
        this.endButtonDiv.style.backgroundImage =
    "url(/sketchbook/images/rightslider.gif)";
        this.endButtonDiv.style.backgroundRepeat = "no-repeat";
        this.timeSliderDiv.appendChild(this.endButtonDiv);
```

```

    // Labels
    this.currentLabelDiv = this._createLabel(-16);
    this.timeSliderDiv.appendChild(this.currentLabelDiv);

    this.startLabelDiv = this._createLabel(21);;
    this.timeSliderDiv.appendChild(this.startLabelDiv);

    this.endLabelDiv = this._createLabel(21);
    this.timeSliderDiv.appendChild(this.endLabelDiv);

    // Update
    this.update();

    // Events
    Event.observe(this.currentButtonDiv, 'mousedown', this.onCurrentButtonDown.bindAsEventListener(this), false);
    Event.observe(this.startButtonDiv, 'mousedown', this.onStartButtonDown.bindAsEventListener(this), false);
    Event.observe(this.endButtonDiv, 'mousedown', this.onEndButtonDown.bindAsEventListener(this), false);
        Event.observe(document, 'mousemove', this.onButtonDrag.bindAsEventListener(this), false);
        Event.observe(document, 'mouseup', this.onButtonUp.bindAsEventListener(this), false);

    },

    /**
     * Converts a time to a slider position (pixels).
     * t time
     * return x
     */
    getPosition: function(t) {
        return this.width / (this.totalInterval.end
            - this.totalInterval.start) * (t-this.totalInterval.start);
    },

    /**
     * Fires a change of interval to all its listeners.
     */
    fireChange: function() {
        // try to notify the listeners
        try {
            for (i=0; i<this.changeListeners.length;i++) {
    this.changeListeners[i].notify(this.currentInterval);
            }
        } catch (err) {
            alert (err);
        }
    },

    /**
     * updates the slider positions.
     */
    update: function() {
        var startPos = this.getPosition(this.currentInterval.start);
```

```
            var endPos = this.getPosition(this.currentInterval.end);
            var currentPos = this.getPosition((this.currentInterval.end + this.currentInterval.start) / 2);

            this.intervalDiv.style.left = startPos + "px";
            this.intervalDiv.style.width = (endPos - startPos) + "px";
            this.startButtonDiv.style.left = startPos - 7 + "px";
            this.endButtonDiv.style.left = endPos + "px";
            this.currentButtonDiv.style.left = currentPos - 7 + "px";

            // Labels
            this.currentLabelDiv.style.left = (currentPos-this.currentLabelDiv.width/2) + "px";
            this.currentLabelDiv.innerHTML = this._format((this.currentInterval.end + this.currentInterval.start) / 2);

            this.startLabelDiv.style.left = (startPos-this.startLabelDiv.width/2) + "px";
            this.startLabelDiv.innerHTML = this._format(this.currentInterval.start);

            this.endLabelDiv.style.left = (endPos-this.startLabelDiv.width/2) + "px";
            this.endLabelDiv.innerHTML = this._format(this.currentInterval.end);
      },

      /**
       * on button down.
       */
      onCurrentButtonDown: function(event) {
            this.dragX = Event.pointerX(event);
            this.dragMode = 1;
            this.button = "current";
      },

      /**
       * on button down.
       */
      onStartButtonDown: function(event) {
            this.dragX = Event.pointerX(event);
            this.dragMode = 1;
            this.button = "start";
      },

      /**
       * on button down.
       */
      onEndButtonDown: function(event) {
            this.dragX = Event.pointerX(event);
            this.dragMode = 1;
            this.button = "end";
      },

      /**
       * on button up.
       */
```

```
	onButtonUp: function(event) {
		if (this.dragMode == 1) {
			this.fireChange();
		}
		this.dragMode = 0;
	},

	/**
	 * on button drag.
	 */
	onButtonDrag: function(event) {
		if (this.dragMode != 1) return;
		var x = Event.pointerX(event);
		var dx = x - this.dragX;
		var dt = dx / this.width * (this.totalInterval.end - this.totalInterval.start);

		if (this.button == "start") {
			this.currentInterval.start += dt;
			if (this.currentInterval.start < this.totalInterval.start) {
				this.currentInterval.start = this.totalInterval.start;
			}
			if (this.currentInterval.start > this.currentInterval.end) {
				this.currentInterval.start = this.currentInterval.end;
			}
		} else if (this.button == "end") {
			this.currentInterval.end += dt;
			if (this.currentInterval.end > this.totalInterval.end) {
				this.currentInterval.end = this.totalInterval.end;
			}
			if (this.currentInterval.end < this.currentInterval.start) {
				this.currentInterval.end = this.currentInterval.start;
			}
		} else if (this.button == "current") {
			this.currentInterval.start += dt;
			this.currentInterval.end += dt;
			if (this.currentInterval.end > this.totalInterval.end) {
				this.currentInterval.end = this.totalInterval.end;
			}
			if (this.currentInterval.start < this.totalInterval.start) {
				this.currentInterval.start = this.totalInterval.start;
			}
			if (this.currentInterval.start > this.totalInterval.end) {
				this.currentInterval.start = this.totalInterval.end;
			}
```

```
            if (this.currentInterval.end < this.totalInterval.start) {
                this.currentInterval.end = this.totalInterval.start;
            }
        }

        this.update();
        this.dragX = x;

        if (this.notifyOnDrag == true) {
            this.fireChange();
        }
    },

    /**
     * Creates a label
     * top y-coordinate
     */
    _createLabel: function(top) {
        var labelDiv = document.createElement("div");
        labelDiv.style.position = "absolute";
        labelDiv.style.overflow = "hidden";
        labelDiv.style.left = "0px";
        labelDiv.style.top = top + "px";
        labelDiv.width = 60;
        labelDiv.style.width = labelDiv.width + "px";
        labelDiv.style.height = "14px";
        labelDiv.style.backgroundColor = "#ffffff";
        labelDiv.style.border = '1px solid #555555';
        labelDiv.style.fontSize = "10px";
        labelDiv.style.fontFamily = "Arial";
        labelDiv.style.textAlign = "center";
        labelDiv.style.zIndex = "1000";
        labelDiv.className = "studio_noselect";
        labelDiv.innerHTML ="";
        return labelDiv;
    },

    /**
     * formats a value.
     */
    _format: function(value) {
        return this.formatter.format(value);
    },

    /**
     * Converts pixels to numbers
     * pixels e.g. 100px
     * returns e.g. 100
     */
    _toNumber : function(pixels) {
        return Number(pixels.substring(0,pixels.length-2));
    }
}
```

A5: Off the shelf libraries

The table below contains a list of off the shelf libraries used in our MPS, see also section 5.3.

Name	Description	URL
DWR	Direct Web Remoting 1.1	http://getahead.org/dwr/
Prototype	JavaScript framework 1.5.0	http://www.prototypejs.org/
VFS	Virtual file system 1.0	http://jakarta.apache.org/commons/vfs/
ROME	RSS/Atom syndication 0.9	https://rome.dev.java.net/
GISBeans	GIS library (version 2004)	http://sourceforge.net/projects/gisbeans/
DSOL	Distributed Simulation Object Library 1.6.6	http://sk-3.tbm.tudelft.nl/simulation/index.php
Java3D	Java 3D	http://java.sun.com/products/java-media/3D/
Jep	Java Expression Parser 2.4.0	http://sourceforge.net/projects/jep/

Appendix B: Map file

Map files are used as input for our *Map* service, see chapter 6. Below we provide an example of a map file. This map file specifies a map with three layers:
- a static background layer (bitmap)
- a dynamic layer showing the transition of an area towards the future (bitmaps)
- a safety contour (shape files)

```
<?xml version="1.0" encoding="UTF-8"?>
<map>
  <name>map</name>
  <units>meters</units>
  <extent>
    <minX>53000</minX>
    <minY>420000</minY>
    <maxX>116000</maxX>
    <maxY>450000</maxY>
  </extent>
  <uri>file:///C:/development/liferay/my_maps/safety_map/</uri>
  <style>
    <color>150,150,150</color>
    <outlineColor>0,0,0</outlineColor>
  </style>
  <size>
    <width>800</width>
    <height>600</height>
  </size>
  <layer>
      <name>backgroundmap</name>
      <data type="raster" folder="static">
            <source width="2100" height="1000" mimeType="jpg">
1-053000_450000</source>
      </data>
      <status>true</status>
      <clickable>false</clickable>
      <!-- An attribute identifying a label -->
      <labelItem>label</labelItem>
      <!-- An attribute identifying a class -->
      <classItem>class</classItem>
      <class>
            <name>land</name>
            <style>
                  <color>123,123,123</color>
                  <outlineColor>0,0,0</outlineColor>
            </style>
            <label>
                  <enabled>false</enabled>
            </label>
      </class>
  </layer>
  <layer>
      <name>dynamic</name>
      <data type="raster" folder="dynamic">
            <source width="800" height="629" mimeType="jpg"
name="currentSituation">
                  <interval>
                        <type>TIME_MOMENT</type>
```

```
                        <start>1/1/2005</start>
                        <end>1/1/2007</end>
                  </interval>
            </source>
            <source width="800" height="629" mimeType="jpg"
name="transitionSituation">
                  <interval>
                        <type>TIME_MOMENT</type>
                        <start>1/1/2007</start>
                        <end>1/1/2015</end>
                  </interval>
            </source>
            <source width="800" height="629" mimeType="jpg"
name="futureSituation">
                  <interval>
                        <type>TIME_MOMENT</type>
                        <start>1/1/2015</start>
                        <end>1/1/2020</end>
                  </interval>
            </source>
      </data>
      <status>true</status>
      <clickable>false</clickable>
      <interval>
            <type>TIME_MOMENT</type>
            <start>1/1/2004</start>
            <end>1/1/2021</end>
      </interval>
      <!-- An attribute identifying a label -->
      <labelItem>name</labelItem>
      <!-- An attribute identifying a class -->
      <classItem>class</classItem>
      <class>
            <name>land</name>
            <style>
                  <color>123,123,123</color>
                  <outlineColor>0,0,0</outlineColor>
            </style>
            <label>
                  <enabled>true</enabled>
                  <type>NORMAL</type>
                  <color>0,0,0</color>
                  <fontName>Arial</fontName>
                  <fontSize>12</fontSize>
            </label>
      </class>
  </layer>
  <layer>
    <name>e10-5</name>
    <data type="shape" folder="safety/e10-5">
      <source mimeType="shp" name="e10-5_1" />
      <source mimeType="shp" name="e10-5_10" />
      <source mimeType="shp" name="e10-5_11" />
      <source mimeType="shp" name="e10-5_12" />
      <source mimeType="shp" name="e10-5_13" />
      <source mimeType="shp" name="e10-5_14" />
      <source mimeType="shp" name="e10-5_15" />
      <source mimeType="shp" name="e10-5_16" />
      <source mimeType="shp" name="e10-5_17" />
      <source mimeType="shp" name="e10-5_18" />
```

```
      <source mimeType="shp" name="e10-5_19" />
      <source mimeType="shp" name="e10-5_2" />
      <source mimeType="shp" name="e10-5_20" />
      <source mimeType="shp" name="e10-5_21" />
      <source mimeType="shp" name="e10-5_22" />
      <source mimeType="shp" name="e10-5_3" />
      <source mimeType="shp" name="e10-5_4" />
      <source mimeType="shp" name="e10-5_5" />
      <source mimeType="shp" name="e10-5_6" />
      <source mimeType="shp" name="e10-5_7" />
      <source mimeType="shp" name="e10-5_8" />
      <source mimeType="shp" name="e10-5_9" />
    </data>
    <status>true</status>
    <clickable>false</clickable>
    <labelItem>name</labelItem>
    <classItem>class</classItem>
    <legend>
        <status>true</status>
    </legend>
    <class>
      <name>e10-5</name>
      <style>
        <color>255,128,0,0</color>
        <outlineColor>255,128,0</outlineColor>
        <width>3</width>
      </style>
      <label />
    </class>
  </layer>
</map>
```

Appendix C: Storyboard APS1 evaluation

C1: test program (English translation)

8:45	Walking in, coffee and thee
9:00	General introduction of the project by Joop Smits
9:10	General introduction of the Area Planning Studio (APS) by Alexander Verbraeck
9:20	Explanation of process support and evaluation matrix (*Matchbox*) in APS by Alexander Verbraeck.
9:30	Introduction of case 'Maasvlakte 2015' by Cees Pons

9:40	Part 1 of the fictive area planning meeting Opening Minutes 12-08-2015 Action list Recapitulation of choices made in the evaluation matrix Selection of land uses based on parcel sizes.
9:55	Reflection on part 1 of the meeting by Alexander Verbraeck
10:00	Coffee and thee break
10:10	Explanation of making available information in the APS by Alexander Verbraeck

10:15	Part 2 of the fictive area planning meeting Possible restrictions with regard to external safety by Cleo Lenger Possible delay due to ground pollution by Cleo Lenger Possibilities for nautical reachablility at the side of the Yangtzeekanaal by Rob Sibbes Demand for space in the port region from the market by Kees van der Hoop Possible restrictions due to too high concentrations of fine dust by door Cleo Lenger

10:45	Reflection on part 2 of the meeting by Alexander Verbraeck
10:50	Coffee and thee break
11:00	Explanation of planning the future using APS by Alexander Verbraeck

11:10	Part 2 of the fictive area planning meeting Possible bottlenecks in road capacity due to expansion of TOR by Cees Pons Overview of already planned projects in the port region by Cees Pons Presence of endangered species by Jasna de Groot Question round Closing the meeting

11:25	Reflection on part 3 of the meeting by Alexander Verbraeck
11:30	Lunch
11:35	During the lunch 10 answering propositions for the discussion/brainstorm session

11:45	Start discussion/brainstorm under supervision of Mariëlle den Hengst
13:00	The end

C2: test program (Dutch)

8:45	Inloop koffie & thee
9:00	algemene inleiding van het project door Joop Smits
9:10	Algemene inleiding van de Area Planning Studio (APS) door Alexander Verbraeck
9:20	Uitleg van procesondersteuning en evaluatiematrix in de APS door Alexander Verbrack
9:30	Inleiding van de casus 'Maasvlakte 2015' door Cees Pons

9:40	Deel 1 van de fictieve gebiedsplannings vergadering Openen Notulen 12-08-2015 Actielijst Recapituleren gemaakte keuzen in de evaluatiematrix Selecteren van invullingen op basis van kavelgrootte
9:55	Reflectie op de deel 1 van de vergadering door Alexander Verbraeck
10:00	Koffie en thee pauze
10:10	Uitleg over ontsluiten van informatie in de APS door Alexander Verbraeck

10:15	Deel 2 van de fictieve gebiedsplannings vergadering Mogelijke beperkingen i.v.m. externe veiligheid door Cleo Lenger Mogelijke vertraging door bodemverontreiniging door Cleo Lenger Mogelijkheden voor nautische ontsluiting aan zijde Yangtzeekanaal door Rob Sibbes Vraag naar ruimte in het havengebied vanuit de markt door Kees van der Hoop Mogelijke beperkingen door te hoge concentraties fijn-stof door Cleo Lenger

10:45	Reflectie op de deel 2 van de vergadering door Alexander Verbraeck
10:50	Koffie en thee pauze
11:00	Uitleg over plannen in de toekomst in de APS door Alexander Verbraeck

11:10	Deel 3 van de fictieve gebiedsplannings vergadering Mogelijke knelpunten in de wegbereikbaarheid bij uitbreiding TOR door Cees Pons Overzicht van de reeds geplande projecten in het gebied door Cees Pons Aanwezigheid van beschermde soorten door Jasna de Groot Rondvraag Sluiten

11:25	Reflectie op de deel 3 van de vergadering door Alexander Verbraeck
11:30	Lunch
11:35	Tijdens de lunch 10 stellingen beantwoorden voor de discussie/brainstorm

11:45	Start discussie/brainstorm begeleid door Mariëlle den Hengst
13:00	Einde

C3: Detailed storyboard (Dutch)

Wanneer	Wie	Wat	Waarom	History	Plot tijdens test	Verwacht resultaat (besluiten en taken)	Producten APS
	Joop Smits	Algemene inleiding, Samenwerking TU-Delft HbR, toename complexiteit en dynamiek, huidige problematiek gebiedsplanning en aanleiding APS-project	Achtergrond van het project duidelijk maken aan de deelnemers.	n.v.t.	-In een aantal slides de achtergrond en ideeen van het APS-project introduceren		
	Alexander Verbraeck	Het concept van de evaluatiematrix uitleggen aan de deelnemers	Deelnemers een overzicht geven van de functionaliteiten die in dit vraagstuk naar voren zullen komen.	n.v.t.	Evaluatiematrix -Doel: een structuur bieden om alternatieve invullingen te te evalueren -Rijen zijn invullingen -Kolommen doelstellingen, criteria	Deelnemers hebben een beeld van de functionaliteiten van de evaluatiematrix en ontsluiting van informatie (GIS, klantgegevens, kavelgegevens)	-Evaluatiematrix - Informatievoorzieningen
	Cees Pons	Inleiding van de casus	De deelnemers laten inleven in de casus en de rol die zij spelen en hebben gespeeld.	n.v.t.	-Het is nu 2015 -Maasvlakte in 2015 laten zien -Euromax is er -Lyondell geeft optie vrij	Deelnemers worden ingeleid in de casus	-kaart
	Cees Pons	Een aantal al gemaakte keuzen in de evaluatiematrix zichtbaar maken en ‘spelen’ met de matrix. De meerwaarde van het kleurgebruik en overzicht zichtbaar maken.	Deelnemers bekend maken met de evaluatiematrix en het en herkennen van relevante ´gaten´ in de matrix.	Evaluatiematrix is deels ingevuld met vastgelegde kleuren, motivatie en informatie over o.a. : -Keuze natte en commerciële uitgifte -Door geluid valt ‘bouwstoffen en mineralen’ af, deepsea container terminal wordt nog wel in beschouwing genomen.	Cees begint met een recapitulatie van de evaluatiematrix van invullingsvormen voor het voormalige Lyondell optie terrein. Een aantal kleine voor de hand liggende handelingen zullen worden gedaan om het idee van de matrix uit te leggen: -Uitleg criteria/doelstelling aanklikken -Openen van een cel zodat achterliggende info & discussies zichtbaar wordt. -Maakt duidelijk dat er verschillende cellen nog moeten worden ingevult en dat vandaag externe veiligheid en nautische bereikbaarheid op de agenda staat.	-Deelnemers hebben een beeld hoe ze de evaluatiematrix moeten gebruiken -Noodzaak om externe veiligheid en nautische bereikbaarheid verder te beschouwen herkennen	Evaluatiematrix
	Cees Pons	Openen vergadering	-	-	-	-	-
	Cees Pons & Saskia Klopstra	Notulen vorige vergadering doorlopen	Geheugen in de organisatie weergeven. Deelnemers echt ‘meenemen’ in de casus	-Notulen van vorige vergadering	-Open notulen door Saskia Klopstra		-notulen

	Cees Pons	Recapituleren van gemaakte beslissingen in de vorige vergadering	Geheugen in de organisatie weergeven. Deelnemers inleiden in de casus	-besluiten van vorige vergadering	-lijst met gemaakte besluiten doorlopen	Goed beeld van de historie van het gebiedsplanningsproces	-besluitenlijst
	Cees Pons & Saskia	Actielijst doorlopen	Ondersteuning proces	-afgevoerde acties ook zichtbaar -nog staande acties	-De actielijst voor deze vergadering tonen, Cees geeft even aan dat de acties in werkelijkheid worden besproken maar dat voor de test slechts één actie zal worden afgevoerd.	Procesondersteuning is besproken	-actielijst
	Cees Pons	Opvragen Matrix en stilstaan bij de mogelijkheid om een deepsea container terminal te plaatsen	Uit te voeren taken en te nemen besluiten kan op basis van 'lege' cellen in de matrix worden bepaald	n.v.t.	-Of een deepsea terminal pas op het terrein is nog niet bepaald. -Cees pons stelt voor om de optie af te laten vallen, immers geen 120ha. -Bart-Luc merkt op dat er in de toekomst wellicht een grotere kavel kan worden uitgegeven wanneer aangrenzende kavels vrij komen.	-Behoefte aan klantinformatie van omringende kavels	-Matrix
	Bart-Luc Olde Hanter	De contractgegevens van omringende kavels opvragen	De mogelijkheid om klantgegevens via de APS beschikbaar te ontsluiten.	n.v.t.	De klantgegevens van omringende kavels worden opgevraagd	Informatie beschikbaar bij het HbR wordt door de APS beter ontsloten	COREN
	Cees Pons & Saskia Klopstra	Concluderen dat de deepsea terminal af kan vallen	Voorleggen aan de deelnemers dat deepsea container terminal moet afvallen	n.v.t.	-voorleggen afvoeren deepsea terminal -besluit nemen -formuleren en vasteleggen in besluitenlijst en matrix	Besluit expliciet maken Vastleggen besluitvorming	-besluitenlijst -matrix
3 min	Alexander Verbraeck	Recapetuleren van de ondersteuning van het proces.	Nut APS benadrukken	n.v.t.	Geheugen in de organisatie (besluiten, notulen) Plannen van proces (taken) Structureren van keuzeproces van invullingen (Matrix) Integrale afweging tussen invullingen mogelijk maken (Overzicht en kleurgebruik in Matrix)	Meerwaarde wordt expliciet	-
5 min	Alexander Verbraeck	Ontsluiten van beschikbare informatie (GIS, COREN, SOCUMAS etc) bij het Hbr	Om deelnemers te laten zien dat met de APS beschikbare informatie uit verschillende bronnen bij het havenbedrijf ontsluit.	n.v.t.	Navigerend door de APS verschillende typen informatie weergeven: GIS, COREN adviezen van specialisten.	Deelnemers zien de mogelijkheid om informatie middelde de APS te ontsluiten.	GIS, COREN, tekstuele adviezen lezen
	Cees Pons	Mogelijke besluiten voorleggen aan deelnemers en aanduiden welke informatie nodig is om	Besluitvorming leiden en inleiden inbreng informatie	n.v.t.	-Lege cel aanwijzen -Woord geven aan specialist in informatie in te brengen -Besluitvorming leiden	Vergadering voorzitten	-Matrix -Agenda -Besluitenlijst -Notulen

		benodigd besluit te nemen					
	Cleo Lenger	Risicocontouren weergeven en de consequenties uiteenzetten voor de mogelijkheden op een deel van de Lyondell optie	Besluitvormers informeren over de risicocontouren op de Lyondell optie.	Taak voor Cleo om de informatie over Externe veiligheid op de Maasvlakte aan te leveren en mogelijke beperkingen op het terrein zichtbaar te maken.	Omdat de 10^{-6} risicocontour halverwege het terrein loop zal bij persoon intensieve activiteiten het groepsrisico kritisch kunnen worden. TOR zal ingelicht moeten worden en zij moeten bij hun terminalontwerp rekening houden met externe veiligheid.	Taak voor Kees van der Hoop om te zorgen dat TOR-line op de hoogte wordt gesteld van de consequenties voor hun ontwerp en peilen op ze nog geïnteresseerd zijn.	-GIS -Matrix
	Cees Pons	Mogelijke besluiten voorleggen aan deelnemers en aanduiden welke informatie nodig is om benodigd besluit te nemen	Besluitvorming leiden en inleiden inbreng informatie	n.v.t.	-Lege cel aanwijzen -Woord geven aan specialist in informatie in te brengen -Besluitvorming leiden	Vergadering voorzitten	-Matrix -Agenda -Besluitenlijst -Notulen
	Cleo Lenger	Schetsen van					
	Cees Pons	Mogelijke besluiten voorleggen aan deelnemers en aanduiden welke informatie nodig is om benodigd besluit te nemen	Besluitvorming leiden en inleiden inbreng informatie	n.v.t.	-Lege cel aanwijzen -Woord geven aan specialist in informatie in te brengen -Besluitvorming leiden	Vergadering voorzitten	-Matrix -Agenda -Besluitenlijst -Notulen
	Cleo Lenger	De bodemverontreiniging meetgegevens zichtbaar maken.	Besluitvormers op de hoogte brengen dat de bodem van de voormalige Lyondell optie onvervuild is.	Taak voor Cleo om de informatie over bodemverontreiniging op de Maasvlakte aan te leveren en mogelijke beperkingen te beschrijven.	Na het tonen van de bodemgegevens blijkt dat er geen sprake is van bodemvervuiling.	Geen vertraging voor het ontwikkelen van het terrein door bodemvervuiling.	GIS
	Cees Pons	Mogelijke besluiten voorleggen aan deelnemers en aanduiden welke informatie nodig is om benodigd besluit te nemen	Besluitvorming leiden en inleiden inbreng informatie	n.v.t.	-Lege cel aanwijzen -Woord geven aan specialist in informatie in te brengen -Besluitvorming leiden	Vergadering voorzitten	-Matrix -Agenda -Besluitenlijst -Notulen
	Rob Sibbes	M.b.v. een tekening de mogelijke nautische ontsluiting via het Yangtzekanaal weergeven.	Communiceren van de mogelijkheden om aan de nautische bereikbaarheids eisen van RKT te voeldoen.	RKT moet schepen kunnen ontvangen tot 19.65 meter. Rob Sibbes heeft de taak gekregen om samen met iemand van Ontwikkeling Infrastructuur te onderzoeken of het mogelijk is om dit type	In de APS moet een bestand worden geopend waarin een tekening te zien dat nautische ontsluiting aan het Yangtzeekanaal de bereikbaarheid van MV-2 beperkt. Aan de Europahaven kan het vaarwater niet worden uitgebaggerd tot de gewenste diepte door de kademuur bij ECT en Maersk.	Nautische eisen kan niet aan worden voldaan	Bestanden uitwisselen Evaluatiematrix

				schepen te ontvangen.	Dit gegeven maakt RKT onmogelijk op deze kavel en daarom zal de cel rood worden gekleurd.		
	Cees Pons	Mogelijke besluiten voorleggen aan deelnemers en aanduiden welke informatie nodig is om benodigd besluit te nemen	Besluitvorming leiden en inleiden inbreng informatie	n.v.t.	-Lege cel aanwijzen -Woord geven aan specialist in informatie in te brengen -Besluitvorming leiden	Vergadering voorzitten	-Matrix -Agenda -Besluitenlijst -Notulen
	Kees van der Hoop	De projectgroep mondeling informeren over de verwachte marktvraag naar havengebied bij de selectie van invullingsvormen in de matrix.	Adv deze informatie kan de projectgroep mogelijk een deel van de matrix invullen.	Kees van der Hoop heeft in de vorige vergadering de marktvraag onderzoeken als taak meegekregen.	Kees van der Hoop informeert de projectgroep over de verwachte duur tot een klant zich wilt vestigen of het tot een klant binnen een generiek marktsegment is gevonden. De projectgroep vult van de invulling selectie de cellen in de kolom 'duur tot ontwikkeling' in de matrix verder in.	Matrix is verder ingevuld waardoor een keuze voor invulling een stap dichterbij is gekomen	Evaluatiematrix
	Cees Pons	Mogelijke besluiten voorleggen aan deelnemers en aanduiden welke informatie nodig is om benodigd besluit te nemen	Besluitvorming leiden en inleiden inbreng informatie	n.v.t.	-Lege cel aanwijzen -Woord geven aan specialist in informatie in te brengen -Besluitvorming leiden	Vergadering voorzitten	-Matrix -Agenda -Besluitenlijst -Notulen
	Cleo Lenger	Schetsen van kritische locaties mbt fijn stof	De additionele fijn stof productie van de TOR-line kan tot problemen leiden.	n.v.t.	-Schetsen van kritische locaties mbt fijn stof -besluiten dat nader onderzoek noodzakelijk is.	-Communiceren van ideeen is mogelijk doormiddel van de schetsfunctie	-schetsen
	Cees Pons	Mogelijke besluiten voorleggen aan deelnemers en aanduiden welke informatie nodig is om benodigd besluit te nemen	Besluitvorming leiden en inleiden inbreng informatie	n.v.t.	-Lege cel aanwijzen -Woord geven aan specialist in informatie in te brengen -Besluitvorming leiden	Vergadering voorzitten	-Matrix -Agenda -Besluitenlijst -Notulen
2 min	Alexander Verbraeck	Recapituleren van de getoonde functies	Benadrukken meerwaarde APS	n.v.t.	-Ontsluiten van beschikbare informatie bij het Havenbedrijf -Besluitvorming structureren adv de matrix -Vastleggen van besluiten in de matrix -Schetsen van ruimtelijke concepten	Meerwaarde wordt expliciet	
5 min	Alexander Verbraeck	De mogelijkheid om ontwikkeling en mogelijke toekomstige	Deelnemers laten ervaren dat de APS de toekomstige	n.v.t.	-Bij de kaart scrollen door de tijd en de ruimtelijke verandering van het gebied zien.	Deelnemers hebben een beter overzicht hoe een gebied er in de toekomst uitziet en kunnen	-Kaart -Evaluatiematrix -Verkeersmodel

		invullingen in de toekomst te plannen en de verwachte effecten van keuzen in de toekomst weer te geven.	ontwikkeling van een gebied zichtbaar kan maken en ondersteuning bied bij het maken van een concreet plan voor verschillende momenten in de toekomst.		-Via de matrix kiezen van een invulling voor een periode in de tijd, door het scrollen naar de juiste periode wordt deze invulling zichtbaar op de kaart. -In het algemeen de mogelijkheden van de effecten van een keuze voor een invullingvorm uitleggen. Toelichten dat voor de casus slechts de verwachte effecten op de toekomstige weginfrastructuur is weergegeven.	een gedetailleerde planning maken van invulling van een gebied. Door de verwachte effecten van keuzen zichtbaar te maken kunnen besluitvormers beter geïnformeerd besluiten nemen.	
	Cees Pons	Uitgaande van het scenario dat TOR uitbreidt de resultaten van de verkeersstudie weergeven.	Mogelijke toekomstige knelpunten in de weginfrastructuur zichtbaar maken voor de deelnemers.	Cees pons heeft voorafgaande aan de vergadering een simulatiestudie gedaan naar de mogelijke knelpunten op de weg.	-TOR als alternatieve invullingmatrix opvragen en zichtbaar maken dat het criterium wegbereikbaarheid nog niet is ingevuld. -Cees pons toont de simulatieresultaten aan de deelnemers en licht deze toe. -Opstellen van taken om een besluit te kunnen nemen om TOR een gewenste invulling is.	-Aan de hand van de verwachte knelpunten zal er een herontwerp moeten worden gemaakt van de directe toegangsweg en de eerste kruispunten, taak Cees Pons. -Een schatting maken van de kosten om de verwachte knelpunten weg te nemen, taak Cees Pons -Kees van de Hoop zal a.d.v. de eerst schatting een eerste business case maken en	-Matrix -Verkeersmodel -Takenlijst
	Cees Pons	Door de tijd scrollen zodat de geplande projecten en ontwikkeling van de Maasvlakte in de tijd zichtbaar wordt.	De deelnemers inzicht geven in de ruimtelijke ontwikkeling van de Maasvlakte.	OTA heeft de tekening van de maasvlakte in 2025 ingebracht voor de 2^{e} vergadering van het gebiedsplanningsteam	-Cees scrolt door de tijd en laat de kaart van 2005 zien. -Cees scrolt door de tijd en laat de kaart van 2025 zien. -De projecten door de tijd worden doorgelopen en groot onderhoud van de Antarticaweg komt naar voren. -Vragen aan Jasna de Groot of dit mogelijk kan worden uitgesteld zodat het budget in de Busines Case van de TOR-scenario kan worden meegenomen	Niet meer één kaar maar een dynamische kaart waarin ook in de tijd kan worden gekeken.	-Kaart -Projectplanning
	Jasna de Groot	Onderhoud van de Antarticaweg toelichten	Mogelijke integratie met herinrichting Antarticaweg als TOR-line uitbreid.	Projectinfo is beschikbaar	-Toelichten dat groot onderhoud kan worden geïntegreerd -Klein onderhoud in 2017 is wel noodzakelijk maar is veel goedkoper	In de actie om een business case te schrijven toevoegen dat rekening moet worden gehouden met extra budget ivm koppeling groot onderhoud.	-Projectplanning -Actielijst
	Jasna de Groot	De aanwezigheid van	Mogelijke vertraging	-Taak Jasna om	-Weergeven GIS laag beschermde	-Vertraagde uitgifte	-Notulen

		beschermde soorten weergeven in de kaart	om actie te ondernemen om de soorten te beschermen.	aanwezigheid beschermde soorten te achterhalen	diersoorten -Vertraging van mogelijke uitgifte inplannen	-Inplannen compensatie, mitigeren, in de notulen weergeven	-GIS
	Cees Pons	Rondvraag	-	-	-	-	-
	Cees Pons & Saskia Klopstra	Herhalen en controleren van de genomen besluiten, opgestelde taken en gemaakte notulen	Het resultaat van de vergadering laten zien. Controleren of alle output is vastgelegd en er geen open eindjes zijn.	n.v.t.	-Doorlopen notulen -doorlopen actielijst -doorlopen besluitenlijst	Geaccordeerde resultaten	-notulen -actielijst -besluitenlijst
	Alexander Verbraeck	Recapituleren van de getoonde functies en mogelijke toekomstige ontwikkeling			-Verkeersmodel -Ontwikkeling in de tijd -Projecten in de tijd	Meerwaarde wordt expliciet	
	Joop Smits	Van een afstand kijken naar de getoonde functies			?		

Appendix D: Questionnaires APS2 evaluation

Vragen over uw functie

Naam (indien niet anoniem):
Functie:
Afdeling:

Ik ben betrokken bij huidige gebiedsplanning project(en).
Ja, Nee, Beetje

1 Inleidende vragen

1.1 Kunt u zich de APS (Area Planning Studio) – testsessie van vorig jaar (19-08-2005) nog goed herinneren?
Ja, Nee, Beetje

1.2 Wat waren in uw beleving de belangrijkste kenmerken van APS?
Open Antwoord

1.3 Wat waren in uw beleving de belangrijkste **positieve** conclusies met betrekking tot de testsessie vorig jaar?
Open Antwoord

1.4 Wat waren in uw beleving de belangrijkste **negatieve** conclusies met betrekking tot de testsessie vorig jaar?
Open Antwoord

1.5 Stelling: APS, zoals gedemonstreerd tijdens de sessie, is een nuttig tool ter ondersteuning van gebiedsplanning.
Helemaal oneens, Oneens, Neutraal, Eens, Helemaal eens, Weet niet

1.6 Denkt u dat APS een nuttig middel kan zijn in ander soortige projecten dan gebiedsplanning?
Ja, Nee, Geen mening

2 Stellingen met betrekking tot het proces

NB:
Efficiëntie: bij efficiëntie wordt aangegeven hoeveel het kost/oplevert om tot een oplossing te komen.
Effectiviteit: de effectiviteit van een oplossing geeft aan hoe goed het probleem wordt opgelost.

2.1 Het huidige gebiedplanning proces verloop efficiënt.
Helemaal oneens, Oneens, Neutraal, Eens, Helemaal eens, Weet niet

2.2 Het huidige gebiedplanning proces verloopt effectief.
Helemaal oneens, Oneens, Neutraal, Eens, Helemaal eens, Weet niet

Hoe denkt u over de volgende stellingen t.o.v. het huidige proces:

2.3 Ik verwacht dat met APS gebiedsplanning efficiënter zal verlopen dan zonder.
Helemaal oneens, Oneens, Neutraal, Eens, Helemaal eens, Weet niet

2.4 Ik verwacht dat met APS gebiedsplanning effectiever zal verlopen dan zonder.
Helemaal oneens, Oneens, Neutraal, Eens, Helemaal eens, Weet niet

2.5 Ik verwacht dat met APS gemaakte keuzes eenvoudiger terug te vinden zijn dan zonder.
Helemaal oneens, Oneens, Neutraal, Eens, Helemaal eens, Weet niet

2.6 Ik verwacht dat met APS taken beter beheerd kunnen worden dan zonder.
Helemaal oneens, Oneens, Neutraal, Eens, Helemaal eens, Weet niet

2.7 Ik verwacht dat met APS gebiedsplannen eenvoudiger en sneller te onderhouden zijn dan zonder.
Helemaal oneens, Oneens, Neutraal, Eens, Helemaal eens, Weet niet

3 Stellingen met betrekking tot het nut van APS

3.1 APS is nuttig voor het correct en consistent presenteren van informatie.
Helemaal oneens, Oneens, Neutraal, Eens, Helemaal eens, Weet niet

3.2 APS is nuttig voor het maken van een integrale afweging van aspecten.
Helemaal oneens, Oneens, Neutraal, Eens, Helemaal eens, Weet niet

3.3 APS is nuttig voor het inzichtelijk maken van de transitie van een gebied naar de toekomst.
Helemaal oneens, Oneens, Neutraal, Eens, Helemaal eens, Weet niet

3.4 APS is nuttig voor het verkrijgen van inzicht in het verloop van het gebiedsplanningproces.
Helemaal oneens, Oneens, Neutraal, Eens, Helemaal eens, Weet niet

3.5 APS is nuttig voor het evenwichter behandelen van de verschillende aspecten.
Helemaal oneens, Oneens, Neutraal, Eens, Helemaal eens, Weet niet

Overige opmerkingen:

Summary

Mainport Planning Suite: software services to support mainport planning.

Sustainable growth and the commercial success of "Mainport Holland", located in one of Europe's most densely populated areas, is threatened by a lack of available land, a congested infrastructure, and an increasingly complex social, economic and political reality. To deal with these threats mainports, such as the Port of Rotterdam, are reengineering their planning processes. Instead of making plans based on an extrapolation of current trends, the aim is now to find answers to what-if questions which are applied to concurrent scenarios. Mainport planning is like solving a large jigsaw puzzle, but unlike a jigsaw puzzle the pieces used to solve the puzzle are not available beforehand, and there is no single best solution. Solving the mainport planning puzzle is a difficult, lengthy, knowledge and information intensive, multi-actor process. The challenge is to support invoking the memory and the creativity of multiple actors, with different objectives, that are specialized in different fields of knowledge, and that work in different contexts, such that effective mainport planning can be conducted.

Adhering to the concept of studios, suites and services as introduced by Keen and Sol[27], we introduced the design of a Mainport Planning Suite (MPS), i.e. a suite of services to support the actors in a studio-based planning process and improve their effectiveness in mainport planning. Following a design science strategy, we designed our MPS in an iterative way. First we studied mainport planning in practice in two exploratory case studies. From the exploratory case studies we found that supporting mainport planning should be focused on visualizing the knowledge and information that is used and produced during a mainport planning process. Based on the principles of visualization found in literature, we constructed a framework that we used to identify technology building blocks and requirements to design and implement an MPS. Based on the identified requirements we chose to design our MPS as a suite of loosely coupled services that are provided by a web portal. The main services provided by our suite are *Map*, *Sketchbook*, *Matchbox*, and *Aspect Explorer*, which support analysis, design, evaluation and choice in mainport planning. These services are supplemented with a number of services that support the process of planning, e.g. action lists, decision lists and project information services.

The MPS was evaluated in two evaluation sessions at the Port of Rotterdam. For each evaluation session a prototype was assembled from the MPS services. During the first evaluation session a team of area planners was invited to participate in a fictive studio-based area planning meeting supported by our suite. Not all functionality was available during first evaluation session; therefore a second evaluation session was organized in which we used structured walkthroughs of an improved prototype. Based on the outcomes of the evaluation sessions, it is our strong conviction that an MPS is potentially useful and usable to improve the effectiveness of studio-based mainport planning. Future research might focus on the integration of simulation models in an MPS, the extent to which an MPS can support the automatic reporting of planning outcomes, the design of an approach for using an MPS, and the extent to which an MPS can be used in spatial planning in general.

Roy Chin

[27] Keen, P.W.G., H.G. Sol (2007). Decision Enhancement Services. Forthcoming.

Samenvatting

Mainport Planning Suite: software services ter ondersteuning van mainport planning.

De duurzame groei en het commerciële succes van "Mainport Holland", dat zich in één van de meest dicht bevolkte gebieden van Europa bevindt, worden bedreigd door een gebrek aan beschikbaar land, een verstopte infrastructuur, en een steeds complexere sociale, economische en politieke werkelijkheid. Om deze bedreigingen het hoofd te bieden zijn mainports, zoals de Haven van Rotterdam bezig met het herontwerpen van hun planningsprocessen. In plaats van het maken van plannen, die op een extrapolatie van huidige trends worden gebaseerd, is het doel nu om antwoorden te vinden op what-if vragen die worden toegepast op alternatieve scenario's. De planning van een mainport is te vergelijken met het oplossen van een grote puzzel, maar in tegenstelling tot een puzzel zijn de stukken die worden gebruikt om het raadsel op te lossen niet vooraf beschikbaar, en is er geen beste oplossing te vinden. Het oplossen van de mainport planningspuzzel is een moeilijk, tijdrovend, kennis- en informatieintensief, multi-actor proces. De uitdaging is het aanspreken van het geheugen en de creativiteit van de betrokken actoren, die verschillende doelstellingen nastreven, in verschillende kennisgebieden gespecialiseerd zijn, en die in verschillende contexten werken, op een dusdanige manier dat mainport planning op een efficiënte manier kan worden beoefend.

Naar aanleiding van het concept van studio's, suites en services zoals dat door Keen en Sol[28] wordt geïntroduceerd, introduceerden wij het ontwerp van een Mainport Planning Suite, MPS, ofwel een suite van de diensten om de actoren in een studio-gebaseerd planningsproces te ondersteunen en hun doeltreffendheid te verbeteren in mainport planning. De MPS is op een iteratieve manier ontworpen volgens de principes van design science. Mainport planning was bestudeerd in de praktijk in twee oriënterende cases. Uit deze oriënterende cases bleek dat het ondersteunen van mainport planning gefocust moet zijn op de visualizatie van kennis en informatie welke gebruikt en geproduceerd wordt tijdens een mainport planning proces. Op basis van de in de literatuur gevonden principes voor visualizatie, werd een framework opgesteld dat gebruikt werd om de technologiebouwstenen en de vereisten te identificeren om een MPS te ontwerpen en te implementeren. Vervolgens werd op basis van de geïdentificeerde vereisten werd ervoor gekozen om de suite te ontwerpen als een suite van loosely coupled services welke beschikbaar gesteld worden via een web portal. De belangrijkste software services van de suite zijn Map, Sketchbook, Matchbox, en Aspect Explorer, welke analyse, ontwerp, evaluatie en het maken van keuzes in mainport planning ondersteunen. Deze diensten werden aangevuld met een aantal diensten die het planningsproces ondersteunen, zoals actielijsten, besluitenlijsten en services voor het bijhouden van projectinformatie.

De MPS was geëvalueerd in twee evaluatiesessies bij het Havenbedrijf Rotterdam. Voor elke evaluatiesessie werd er een prototype geassembleerd van de MPS software services. Tijdens de eerste evaluatiesessie werd een team van gebiedsontwerpers verzocht om aan een fictieve, studio-gebaseerde gebiedsplanningsvergadering deel te nemen die door de MPS werd ondersteund. Niet alle functionaliteit was beschikbaar tijdens eerste evaluatiesessie, daarom werd er een tweede evaluatiesessie georganiseerd waarin een verbeterd prototype werd geëvalueerd door middel van een structured walkthrough. Op basis van de resultaten van de

[28] Keen, P.W.G., H.G. Sol (2007). Decision Enhancement Services. Forthcoming.

evaluatiesessies, is het onze sterke overtuiging dat een MPS potentieel nuttig en bruikbaar is om de effectiviteit van studio-based mainport planning te verbeteren. Voorstellen voor vervolgonderzoek omvatten de integratie van simulatiemodellen in een MPS, de mate waarin een MPS in staat is om automatisch planningsresultaten te presenteren, het ontwerp van een methode voor het gebruik van een MPS, en de mate waarin een MPS in ruimtelijke planningvorming in algemeen gebruikt kan worden.

Roy Chin

Curriculum Vitae

Roy T.H. Chin was born on July 22nd, 1975 in Leiderdorp, The Netherlands. After graduating from the Maerlant Lyceum in The Hague in 1993 he studied Aerospace Engineering at the Delft University of Technology (1993-1999). During his studies he had several jobs as a student assistant. His master's thesis was titled MIFAS, a modular interactive fast-time Air Traffic Control simulator. After getting his master's degree he worked for several years as a researcher at the Delft University of Technology. At the Faculty of Aerospace Engineering he conducted several projects in the field of air traffic control, aircraft noise, airport business, aero-elasticity of wind turbines, and software development. He also had the opportunity to participate in a course in business planning provided by the TU Delft and Arthur D. Little. His team was rewarded with a second place in the race for best business plan. In 2003 he started his Ph.D. research at Systems Engineering, a section of the Faculty of Technology, Policy and Management. The result of this research is presented in this thesis. Part of this work has been published in several articles and presented at international conferences. Furthermore he has been actively involved in education. He organized, coordinated and taught several lab exercises, taught a course in Visual Basic programming, and supervised several students during their master's thesis project.