Experimental Evaluation of an Object-Oriented Modeling Method: Designing a Multimedia System

Mikael B. Skov and Jan Stage
Department of Computer Science
Aalborg University
Fredrik Bajers Vej 7
DK-9220 Aalborg East
Denmark
Email: dubois@cs.auc.dk / jans@cs.auc.dk

Abstract: New technologies challenge our methods for system analysis and design. This article reports from an experimental evaluation of a typical object-oriented modeling method. The method was used to design a multimedia system that simulates a certain real world environment in order to train and assess the decision-making capabilities of persons operating in that specific setting. Based on an empirical study of this design process, we emphasize strengths and weaknesses of the modeling method. In addition, the article reports from an empirical study of a process where general knowledge about story telling and filmmaking was used as the methodological basis for designing a similar multimedia system. From this second study, we elicit ideas for increasing the extent to which the object-oriented modeling method can support the design of a training and assessment multimedia system.

Keywords: Evaluation, design methodologies, multimedia

1 INTRODUCTION

Methods for analysis and design of information systems inherently relate to specific computer technologies. For example, the structured analysis and design methods were basically influenced by the characteristics of the mainframe technology. Similarly, the object-oriented approach embodies key concepts of the graphical user interface and client server technologies (Bennett et al. 1999).

This inherent relation between method and technology implies that our established ways of conducting system analysis and design are challenged when new technologies emerge. We cannot expect that a method for design within one technology can be adapted without problems to a different technology.

In the mobile millennium, we face complicated tasks of understanding and designing information systems without reference to a specific geographical location. Many systems will be required to support the work of their users while they are in different locations or on the move. The stream of small mobile devices for a multitude of purposes represents one way of handling the requirement to mobility. The tendency here is that the size of the device is decreasing while the required interaction between device and user is increasing. These opposite forces make it virtually impossible to develop a good screen design, because it must be capable of displaying a lot of information while being so small that it is nearly impossible to see anything on it.

The contradictory requirements to mobile systems can be resolved by introducing more media. The term multimedia has been used to denote systems with two fundamental characteristics. First, multimedia systems are characterized by an intensive interaction between the user and the system (Skov and Stage 2000). Second, multimedia systems integrate several modalities, e.g. sound, text,
video, graphics, and animation. The term multimedia means 'multiple media', and the term modality is simply introduced to increase precision, so this characteristic is inherent in the term (Pauen et al. 1998, Wilson 1998). Russo (2000) continues by saying that multimedia systems are more than just nice looking screens; they integrate complex multimedia structures.

Research in multimedia system development shows that contemporary multimedia systems are designed and created primarily by intuition. No methodological support is applied, and projects are characterized by very unsystematic work practices (Sutcliffe and Faraday 1994, 1997). It seems as if this category of system development projects completely ignore the body of knowledge that has been established in software engineering and information system development. A fundamental lesson learned through many studies and experiments in software engineering is that improvements in analysis and design processes require systematic work practices that involve well-founded methods (Fairley 1985, Pressman 1996, Sommerville 1992). Nevertheless, multimedia systems are often designed in an ad-hoc manner (Pauen et al. 1998).

This article explores whether the apparent separation between system analysis and design methods and multimedia design practice is caused by deficiencies of the methods. The aim is to identify the extent to which a typical object-oriented modeling method can support the process of designing a multimedia system. In addition, we describe the work practice that has been applied in the design of multimedia system. The multimedia systems in both cases are for training and assessing the decision-making capabilities of persons operating in a specific context.

In section 2 we start by defining the basic characteristics of designing multimedia system. In addition, we describe the class of multimedia systems for training and assessment. Section 3 gives an overview of the empirical study that provides the foundation of the article. Section 4 describes the process of designing a multimedia system based on an object-oriented modeling method. Section 5 provides ideas for improving the weaknesses of the object-oriented method. Finally, section 6 concludes the article and discusses ideas for further research.

2 DESIGN OF TRAINING AND ASSESSMENT MULTIMEDIA SYSTEMS

1 Training and Assessment Multimedia Systems

The class of multimedia systems is often difficult to define and delimit since it involves many different kinds of computer-based systems. As illustrated above, more people apply a technological definition on multimedia systems stating that multimedia systems describe a computer system integrating more media forms like text, graphics, audio, full-motion video etc., cf. (Hemsley 1997, Pauen et al. 1998, Wilson 1998). In this sense, a multimedia system could be any kind of computer-based system with more than one modality. Donaldson and Cowderoy (1997) define multimedia systems as an experience that has resulted from the convergence of the entertainment, telecommunications, and computing. We rely on combination of both definitions where multimedia systems integrate more mediums or modalities in the interaction with the user, but multimedia systems are also computer-based systems that e.g. have to entertain or to train the user.

Training and assessment multimedia systems are multimedia systems intended for training people’s skills within a given field of expertise or assessing peoples skills or knowledge also within a given field. Training and assessment multimedia systems often integrate several modalities and also apply full-motion video to illustrate real-life situations in which the user has to be trained or assessed. The idea is to confront the user with different situations where decisions or actions are to be made. Often the technology used for these systems are desktop computers with ordinary input and output devices. As an illustrative example, take a multimedia system for training coming automobile drivers. The objective is to train drivers in evaluating and make decisions in certain specific traffic situations when driving an automobile but not to actually train the driver in practical control of the automobile. Such a system will probably involve different modalities such as video-clips from drives, sounds from the engine and other road users, speak from passengers in the car, animation of instruments etc. The
purpose of integrating these modalities is to create a setting that gives the driver a realistic and natural experience. This feature is sometimes referred to as immersion, meaning that the user experience it as if he or she is immersed into the real situation. The system is highly interactive because the driver will constantly use different instruments in the car, and the system will have to respond by presenting the resulting situation. The system involves several modalities and a key problem is to integrate and synchronize them in order to make the experience coherent.

2 What is Design?

Booch (1994) defines design as the disciplined approach one uses to invent a solution for some problem, which also provides a path from requirements to implementation. Jacobson et.al. (1999) describe a model as an abstraction of a system specifying the modeled system from a certain viewpoint and at a certain level of abstraction where a viewpoint could be the design of the system. One can then argue that the design can be described as being the foundation and the specification for the implementation of the future computerized system.

We will apply a pragmatic interpretation of what constitutes a design. We argue that a successful design is a design that provides a good and solid foundation for the implementation of the system. That is the design of training and assessment multimedia systems has to provide the designers with a good and adequate basis for the programming of the system. That is, at this point all major parts of the system has been analyzed and described and requirements for the system are met by the content and the functionality of the system.

Based on these observations, we have identified five key components for the design of training and assessment multimedia systems as illustrated in figure 1.

<table>
<thead>
<tr>
<th>Design Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Model specification</td>
<td>The system will model aspects of the future users use</td>
</tr>
<tr>
<td>2. Functional requirements specification</td>
<td>This refers to the fact that the user has to interact with the system, how this is done and which functions are made available to the user.</td>
</tr>
<tr>
<td>3. User interface design</td>
<td>Design or outlines of the user interface including the structure of all interface screens and of each individual screen layout.</td>
</tr>
<tr>
<td>4. System modeled situations specification and design</td>
<td>The system will define and present a number of real-life situations to the user. These situations have to be identified, designed, and specified.</td>
</tr>
<tr>
<td>5. Overall story and assets design and production</td>
<td>The situations may be a part of a larger story where certain situations have to be showed in a specific order or where causality exists between some situations. Also, it should be decided and designed what assets should</td>
</tr>
</tbody>
</table>

Figure 1. Design components for training and assessment multimedia systems

3 EMPIRICAL STUDY

We have studied two multimedia design processes in order to explore the question raised in the introduction. The systems designed in these processes are denoted as the Umpire System and the Manager System. Our studies were descriptive, in vitro observation involving expert participants, and they were conducted as qualitative case studies, since the variable scope was not defined a priori (Basili 1996). Below, it is described how each study was conducted.
3 The Umpire System

Setting: The process of designing the Umpire System was based on a general object-oriented analysis and design method (Mathiassen et al. 2000). The purpose of this study was to evaluate the relevance of a typical software method when designing multimedia systems; thus the guidelines of the method were strictly obeyed throughout the process, and it was documented to what extent the method provided relevant support. The design process was started in November 1997 and ended in February 1998.

Participants: The team that designed and implemented the Umpire System consisted of a designer and two programmers, all of them with a computer science background. The designer, who is one of the authors of this article, had several years of experience with object-oriented analysis and design methodologies. He was also a match racing umpire and, thereby, was able to act as an expert user. The two programmers had previous experience both with the object-oriented analysis and design methodology and various implementation tools.

Data Collection: During design and implementation, all three participants kept diaries on a daily basis. The diaries describe the work done and the extent to which it was supported by the analysis and design method. In addition, all versions of the analysis and design documents were saved. This documentation is publicly available on the World Wide Web (reference removed for the purpose of blind-refereeing).

4 The Manager System

Setting: The Manager System was designed and implemented for a management consultant company that assists a broad group of large companies in hiring middle-level managers. The study of this process served to document a successful multimedia system design, and thereby provide ideas for solving potential shortcomings of the software design methodology. The design process in question was considered successful in the sense that the consultant company, after being presented with the prototype, decided to finance the development of a fully operational version of the system. The design team did not use a specific method. Instead, narrative knowledge and story-telling techniques (Galyean 1995, Laurel 1993) were employed as a general methodological foundation. The design process was started in January 1997 and ended in December 1997.

Participants: The team that designed and implemented the Manager System consisted of three designers with an education in humanistic computer science; an education where a general humanistic background is combined with selected computer science topics. The humanistic background includes communication, psychology, narration, and media production. In addition, one of them had previously worked in the software industry as a multimedia developer, and the two others had worked with traditional media production. None of them had any prior knowledge of the application domain of the multimedia system that was designed.

Data Collection: The empirical study of this design process started with a semi-structured interview of the three members of the design team. The interview was conducted by two other persons, including one of the authors of this article, and it was made right before the implementation of the prototype was finished. It was based on a checklist with a number of overall topics combined with an open-ended approach where the aim was to allow the participants to use their own vocabulary in order to enable new topics of interest to emerge. The interview lasted approximately one and a half hour and it was fully transcribed into a document of 21 pages. This document is publicly available on the World Wide Web (reference removed for the purpose of blind-refereeing). In addition, the design documentation was studied, and the design team answered further questions in a couple of short, informal sessions.

5 Data Analysis

Each design process was initially described in terms of the phases it comprised. A phase is a limited period of time that ends with a product that can be identified. A single phase may involve many activities with different characteristics, including analytical, evaluating, and constructive efforts, that
all contribute to the product of the phase (Andersen et al. 1990). For each phase it was described how it was conducted and which methodological problems the design team encountered.

The efforts in the design process that employed the object-oriented method were then related to the design issues presented in the previous section. This enabled us to identify the extent to which design issues were actually dealt with in the design process. A similar analysis has been made with the design process that was based on general narrative knowledge, with the purpose of highlighting areas where this approach might provide useful ideas for extending the software design approach.

The results of this analysis is presented in the following two sections. The specific design of the empirical study imposes a number of limitations on these results. First, a study of two design processes cannot provide a valid basis for definite quantitative conclusions. Instead, we have adopted a qualitative approach with the aim of identifying problems and providing ideas for solutions. Second, the teams that conducted the design processes were very different; and three, both processes lead to implementation of a prototype as opposed of a fully implemented system. In order to handle the last two limitations, we determined methodological success in our qualitative study as lack of problems encountered in the process and a resulting design that served as a suitable basis for implementation.

4 THE UMPIRE SYSTEM

The aim of the Umpire System Project was to design a multimedia system that was useful for training and assessing the decision-making qualifications of an umpire. The idea was to exploit multimedia technology to create a situation close to real on the water judging in which the individual umpire could be better trained and more effectively assessed. Match race is a special kind of sailing where only two boats race against each other. During a race umpires decide disputes between two competitors immediately. This approach to racing significantly reduces the lengthy and straining protest hearings that are often experienced in conventional sailing regattas. Yet it also imposes strong requirements on the umpires in terms of their ability to observe and judge almost immediately in a developing situation by applying a complicated set of rules.

The Umpire System Project was carried out exactly as prescribed by the object-oriented analysis and design method OOA&D, cf. (Mathiassen et.al. 1997, 2000). This method combines the strengths of three widely known methods: the object-oriented concepts from (Booch 1994), the relation between real world objects and the dynamic model inside the computer system from (Jackson 1983), and the practical guidelines for object-oriented analysis and design from (Rumbaugh 1991). A brief summary of the analysis activities of OOA&D can be found in (Mathiassen 1995).

The design process in the Umpire System Project consisted primarily of four major phases namely system definition, problem domain analysis, model component design, and the first prototype (see also appendix). The purpose of the system definition phase was to get a preliminary overview of the project. This phase described three fundamental elements. First, the problem domain which is the part of reality that the computer system is used to administer, monitor, or control. Second, the application domain which is the organizational unit that uses the computer system to support their work. Thirdly, the computer system that is a collection of software components that maintains a dynamic model of the problem domain in order to provide relevant information about this domain to the users in the application domain. The purpose of the analysis phase was to specify requirements in terms of these fundamental elements. Most of the effort focused on the problem domain. The application domain of the umpire system is very simple as it is one user simulating a specific and specialized task by means of a stand-alone system. Thus, this part of the requirements was only outlined.

After the requirement specification, the designer started to identify and specify the classes of the system. First, it was decided that the system should contain classes of the users of the system and the calls made by these users since there was a need for registering user actions and decisions. These actions would serve for later evaluations of the user’s umpiring activity. Second, it was also decided that the system should integrate classes of the various match race situations. During match race, the term situation is used for defining a number of events leading to a protest from one or both boats. However, initially the term situation was used to define a sequence of protests and each single protest from the boats was called protest. It was chosen to specify a class called situation defining a match
race situation leading to a protest. Another class called course, which defines more situations related to a specific part of match race, aggregates this class. This modeling activity resulted in the class diagram illustrated in figure 2a.

![Class Diagram](image)

**Figure 2.** (a) The class diagram for the Umpire System. (b) The behaviors of the Situation class.

The modeling of behaviors for the classes caused some problems. The behaviors for the situation class were actually quite simple since no user actions or inputs would cause any changes or updates in the classes. In hand, the designer found it difficult to distinguish between the behaviors of classes modeling the situations of a match race and the behaviors of the classes modeling the users usage of the system. Furthermore, objects of the situation classes cannot be created, updated, or deleted by the user and hence the creation and deletion of these should not be modeled. As illustrated in figure 2b, these objects were designed to start in a passive state until they were actually played for the user in which they would become active.

The design phase was initiated with an analysis of the risks of the project. This analysis is an integrated activity of the method and it is used as a means to understand and manage the design process. The rest of this design phase mainly focused on the model component. There was too limited knowledge of the development tool to make any architectural design decisions, and the analysis results produced so far only facilitated an overall outline of the user interface component.

The first prototype was implemented in order to explore the extent to which a realistic use situation could be created and to get a better understanding of the architecture of the implementation tool as well as the application that are built with it. The first prototype contains one match race situation that shows the maneuvering of two boats over a period of about one minute and ends with a protest from one of the boats. This situation is basically made as a video clip, being recorded from the position that an umpire boat would typically have in that situation, combined with two soundtracks containing the voice of a fellow umpire and a wing observer. The prototype shows that it is possible to implement the requirements to the system, and that this design creates an acceptable use situation.

Summarized, it can be concluded that the application of the object-oriented method for design of a training and assessment multimedia system had the following characteristics:
<table>
<thead>
<tr>
<th>Design Component</th>
<th>The Umpire Systems Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model specification</td>
<td>Good support of design of the model. Strong support for structuring and classification of the software components.</td>
</tr>
<tr>
<td>Functional requirements specification</td>
<td>Good support. However, the application domain analysis gave only little knowledge of the system in use and hence ideas to functions to the system.</td>
</tr>
<tr>
<td>User interface design</td>
<td>Systematic support for designing the user interface.</td>
</tr>
<tr>
<td>System modeled situations specification and design</td>
<td>Problems in identifying and describing situations. Also, too simple descriptions of situations since no user action will update or delete information related to situations.</td>
</tr>
<tr>
<td>Overall story and assets design and production</td>
<td>No means for design and production of the overall story in the system. Also, no means for design of assets.</td>
</tr>
</tbody>
</table>

Figure 3. Design components for the Umpire Systems Project

5 THE MANAGER SYSTEM

The aim of the Manager System Project was to design and develop an assessment and training multimedia system for a consultant company. The system should assist business consultants in the process of assessing and selecting middle-level managers for open positions. The basic idea was to simulate typical work situations and compel an applicant to make decisions and manage in these situations. Assessing and selecting good managers for open positions is a difficult task. In addition to technical skills, one must also assess personal qualifications regarding staff management, negotiation, planning, scheduling etc. The usual practice is to assess and train managers by means of various tests such as questionnaires and case scenarios in combination with different kinds of interviews.

The Manager System Project was fundamentally explorative in its approach to system development. The key techniques employed were founded in narrative theory originating from the humanities and included story writing and movie making, cf. (Galyean 1998) (Laural 1993).

The main phases of the project were: prototyping, story creation, and implementation (see also appendix). The prototyping phase had two distinct purposes. First, it enabled the designers to evaluate the usability of the implementation platform. Second, it gave the customers of the project "proof-of-concept". The prototype was developed within a week by means of an author tool, and it can be characterized as an experimental prototype, cf. (Floyd 1984); it incorporated basic functionality and depicted work situations by means of photos. The prototype was demonstrated to the consultancy company managers who accepted and decided to support the idea of using a multimedia system in their business. After the demonstration, the prototype was discarded.
The story creation phase produced the interactive story with which the prospective users of the system would be confronted. Two activities were dominant within this phase: scene description and act description. Conceptually a story is made up of a number of acts that each consists of a number of scenes. At the micro level, the designers described scenes by means of textual descriptions, on average two pages, including descriptions of cast members, their roles and lines, the environment, the duration of the scene as shown in the upper left corner of figure 4. At the macro level a script was created to describe the flow of the acts that tied the individuals scenes together. The acts were visualized by the use of simple state transition diagrams depicting the flow of the interactive story in terms of sequence and selection. Each state represented a situation in which users were to make a decision. Through an iterative process, the acts and scenes were joined into a coherent interactive story. The scene descriptions were the main foundation of the film shooting that produced the movie parts, whereas the act descriptions were the foundation of the media integration phase, where the system was eventually built. In figure 4, this is illustrated by textual descriptions of scenes, their implementation as video clips, and the overall structure between these video clips and user decisions. The dark boxes and arrows represent a followed path by one user of the system.

The tool enhancement phase involved the construction of an enhanced implementation platform. In the system architecture design it was decided that progress the project re-implementation. The purpose of this phase was to enable designers to easily required the ability to easily modify the implementation of the story within the tool; there was a lot of generality in story creation with regard of integrating media elements with the story and in specifying when and how users could interact. A market survey did not reveal appropriate tools for this, and it was decided that the project should design its own dedicated author tool. The tool was then used in the final media integration phase to build the interactive stories within the system.

Summarized, it can be concluded that the application of general narrative knowledge for the design of a training and assessment multimedia system had the characteristics listed in figure 5. Also, as an overall weakness of the project since the technical system was not documented the design team was very dependent on individual programming skills.
<table>
<thead>
<tr>
<th>Design Component</th>
<th>The Manager Systems Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model specification</td>
<td>No explicit support of design of the model. However, the model in this project was fairly simple. Low robustness of the system due to lack of control of changes in the technical system.</td>
</tr>
<tr>
<td>Functional requirements specification</td>
<td>No explicit support for the design of functional requirements.</td>
</tr>
<tr>
<td>User interface design</td>
<td>No techniques or means provided which lead to arbitrary design of the user interface. However, the user interface was coherent and standardized.</td>
</tr>
<tr>
<td>System modeled situations specification and design</td>
<td>The creation of scenes and acts made it possible for the designers to continuously evaluate the realism of the system in use.</td>
</tr>
<tr>
<td>Overall story and assets design and production</td>
<td>Good and solid support for the design and creation of the storytelling part of the system. The shooting (and hence creation of assets) of scenes could be done in one go.</td>
</tr>
</tbody>
</table>

Figure 5. Design components for the Manager Systems Project

6 CONCLUSION

Multimedia system design processes should rest on a solid methodological foundation. This, however, seems to contradict the creative but unstructured and casual atmosphere that characterizes the work practices of many contemporary efforts in multimedia system development. The multimedia literature does include proposals for methods, for example Donaldson and Cowderoy (1997), England and Finney (1999), Hemsley (1997), and Sutcliffe and Faraday (1994, 1997), but they are notable exceptions to the contemporary focus on the capabilities of multimedia and the related technical challenges.

This article reports from an empirical study of an experiment where a typical object-oriented modeling method was used to design a multimedia system for training and assessing the decision-making capabilities of persons operating in a specific environment. The method facilitated a clear description of the problem domain, the application domain, and the purpose of the computer system. It also supported the designer in producing a coherent and understandable description of the problem domain classes and their structural relationships. A number of weaknesses of the object-oriented method were also identified. It was difficult to distinguish the dynamics of the situation objects that are embedded in the system in order to simulate real world situations from the dynamics of the objects that represent the user’s activities when using the system. Furthermore, the application of use-cases facilitated no improved understanding of the use situation and the application domain. Finally, the production of the assets that are used to create the realistic experience and their integration into the software system was not supported by the method and, therefore, it was more or less neglected in the design process. In conclusion, the object-oriented method did support the process of designing a training and assessment multimedia system by maintaining a systematic focus on important technical properties but failed to address the heart of the system.

The article also reports from an empirical study of a design process based on general narrative knowledge. The main strength of this approach was its focus on the scenes and acts that prospective users of the system will experience. The use of scripts and scene descriptions combined with state transition diagrams for visualizing the interactive story made it possible for the designers to ensure that the interactive story was coherent, to continuously evaluate the realism of the system in-use, and to shoot the video footage in one session. The main weakness of this approach was that the design
process was more dependent on individual skills of the designers, rather than a methodological
founded design. The design of the technical system is not documented, thus there was a risk that the
project could have failed, due to lack of system design documentation. Furthermore, the use of state
transition diagrams to visualize the story was selected without knowledge of state transition diagrams;
it was a pragmatic decision that should enable description of an interactive story without introducing
too much complexity. The low emphasis on the software system to be developed implied that the
available technical platform had to be enhanced, and the prototype suffered from a low degree of
robustness. In conclusion, this design process addressed the heart of the system but in a less structured
manner and only with limited emphasis on technical issues.

From the empirical studies of the two design processes, we can emphasize the following lessons
learned:

1. The construction of a multimedia system is inherently a software design process.
2. Narrative structures are essential for the design of training and assessment multimedia systems.
3. Production and use of assets has to be supported in the design process.

These lessons can be seen as fundamental requirements for a modeling method for multimedia
systems. It should be emphasized that the empirical foundation for these conclusions are two limited
design processes where the size of the problem and applications domains are quite small. One
immediate consequence is that techniques to handle complexity, which is a strength of object-
orientation, have not been vital to the design process. Therefore, experiments involving larger software
systems might be more favorable to the object-oriented approach.

The research in this paper has revealed two potential avenues for further development in this field.
First, a combination of the two approaches may seem to be an obvious way for further improvement in
the field of multimedia design. Yet it is by no means trivial to suggest how this combination should be
made. Second, a more ambitious approach would be to develop new concepts and descriptions that on
the one hand involve general narrative knowledge and on the other hand relates clearly to a sound
technical implementation. This problem has been solved in hypermedia applications by introducing
new types of classes, e.g. classes for describing narrative structures (Pauen et al. 1998)

ACKNOWLEDGEMENTS

The research behind this article has been partially financed by the Danish Natural Science Research
Council, programme No. 11-8394, and the Danish Research Councils’ joint Multimedia Programme
No. 960869. Steen Andersen and Peter Stegenborg Larsen did the implementation of the Umpire
System prototype, and Ian Peter Seme, Gabriel Schoenau Hansen, and Claus Rosenstand developed
the Manager System. We are very grateful for their contributions.

REFERENCES

Andersen, N. E., Kensing, F., Lassen, M., Lundin, J., Mathiassen, L., Munk-Madsen, A., and
Hall.

Proceedings of the 18th International Conference on Software Engineering. pp. 442-449. IEEE
Computer Society Press, Los Alamitos, California.


Benjamin/Cummings.

and J. McMullan (Eds.), Achieving Software product Quality, pp. 189-205.
APPENDIX

The following two figures illustrate the phases and the products of the two training and assessment multimedia systems design processes.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Definition</td>
<td>System definition</td>
</tr>
<tr>
<td></td>
<td>• Specification of the problem domain, application domain and the purpose of the computer system</td>
</tr>
<tr>
<td>Problem Domain Analysis</td>
<td>Analysis document (13 pages):</td>
</tr>
<tr>
<td></td>
<td>• Complete model of problem domain classes, their structural relationships, and their dynamic behavior</td>
</tr>
<tr>
<td></td>
<td>• Overall description of application domain, functionality and user interface</td>
</tr>
<tr>
<td>Model Component</td>
<td>Project risk analysis</td>
</tr>
<tr>
<td></td>
<td>Design document (7 pages):</td>
</tr>
</tbody>
</table>
Design
• Detailed specification of the model component
• Outline of the user interface component
Revised analysis document

First Prototype
A prototype with:
• Full functionality in the primary use situation
• Video representation of one situation

**Figure A1**: Phases and products in the design of the Umpire System.

<table>
<thead>
<tr>
<th>Design Component</th>
<th>The Manager Systems Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model specification</td>
<td>No explicit support of design of the model. However, the model in this project was fairly simple. Low robustness of the system due to lack of control of changes in the technical system.</td>
</tr>
<tr>
<td>Functional requirements</td>
<td>No explicit support for the design of functional requirements.</td>
</tr>
<tr>
<td>specification</td>
<td></td>
</tr>
<tr>
<td>User interface design</td>
<td>No techniques or means provided which lead to arbitrary design of the user interface. However, the user interface was coherent and standardized.</td>
</tr>
<tr>
<td>System modeled situations</td>
<td>The creation of scenes and acts made it possible for the designers to continuously evaluate the realism of the system in use</td>
</tr>
<tr>
<td>specification and design</td>
<td></td>
</tr>
<tr>
<td>Overall story and assets</td>
<td>Good and solid support for the design and creation of the storytelling part of the system. The shooting (and hence creation of assets) of scenes could be done in one go.</td>
</tr>
<tr>
<td>design and production</td>
<td></td>
</tr>
</tbody>
</table>

**Figure A2**: Phases and products in the design of the Manager System.