



Designing interactive narrative systems: is object-orientation useful?

Mikael B. Skov*, Jan Stage

Department of Computer Science, Aalborg University, Fredrik Bajers Vej 7, DK-9220 Aalborg East, Denmark

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 that was used to design an interactive narrative 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 this empirical study, we emphasize strengths and weaknesses of the object-oriented method. To complement this evaluation, the article also reports from a related empirical study of a process where general knowledge about storytelling and filmmaking was used as the methodological basis for designing a comparable 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 system. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Interactive narratives; Storytelling; Evaluation; Design methodologies; Object-oriented analysis and design

1. Introduction

Interactive narrative system is being used to denote a new kind of computerized system that enable users to create their own stories and experience while using the system [1]. In traditional narrative forms, e.g. written novels or movies, the discourse of the storytelling is determined in advance by the author. In interactive narrative systems, the discourse is not completely predefined but emerges as a result of the interaction between the user and the system, and the user's ability to influence the discourse is by selecting different paths through the narrative [2]. Design, implementation, and use of interactive narrative systems have been reported from such diverse areas as entertainment [1], illness treatment [3], training of collaboration skills [4], and assessment of competence [5].

The class of interactive narrative systems is difficult to define and delimit both because of the diversity of systems and the rapid technological developments that

constantly move the boundaries of what we are able to implement. In this article, we will use the term interactive narrative systems, or just interactive narratives, to denote systems with four fundamental characteristics that are inspired by [6–9]. First, they integrate several elements based on various modalities, e.g. sound, text, video, graphics, and animation. Second, they integrate complex structures to control and synchronize the individual elements. Third, they are characterized by an intensive interaction between the user and the system. Fourth, they are based on a narrative structure that regulates how the use of the system can develop over time. The first two characteristics are consistent with simple definitions of multimedia systems. The last two characteristics are introduced to emphasize the combination of storytelling and user activity. The four characteristics also explain why the class of interactive narrative systems has been described as being considerably more functionally complex than traditional interactive systems [10].

The characteristics of interactive narratives and their inherent complexities may also influence the development process. It seems reasonable to expect that

*Corresponding author.

E-mail address: dubois@cs.auc.dk (M.B. Skov).

development of systems with such qualities demand highly structured design and implementation processes. However, this expectation is not confirmed by studies of practice. Research shows that contemporary interactive narrative systems are designed and created primarily by intuition. No methodological support is applied, and projects are characterized by very unsystematic work practices [11,12]. It seems as if this category of system development projects completely ignores 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 these areas is that improvements in analysis and design processes require systematic work practices that involve well-founded methods [13–15]. Attempts to establish a better methodological foundation are occurring, e.g. [16]. There are also a few examples of methods for developing interactive narrative systems, but they tend to focus on certain aspects of such systems, e.g. [11,12,17]. So presently, interactive narrative systems are usually designed in an ad-hoc manner [6].

This article explores what the process of designing an interactive narrative system can learn from an established software engineering tradition. In Section 2, we present two experiments that form the empirical foundation of the article. This also includes a description of the specific sub-category of interactive narrative systems that we are dealing with. We denote this category as training and assessment systems, because they are used for training and assessing the decision-making capabilities of persons who are operating in a specific context. Section 3 describes how a contemporary software engineering method based on the object-oriented approach can be used to design a training and assessment system. This description is based on an empirical study where an object-oriented modeling method was used to support analysis and design of an interactive narrative system. This experiment leads to identification of strengths and weakness of the method that was employed. Section 4 provides ideas for improving the weaknesses of the object-oriented method through analysis of a development project that produced a successful training and assessment system. These ideas are based on a second empirical study. Finally, Section 5 concludes the article and discusses ideas for further research.

2. Empirical study

We have studied two interactive narrative system design processes in order to explore the questions raised in the introduction. Our studies of these processes 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 [18]. Below, we describe the category of interactive narrative system we have studied.

2.1. Training and assessment systems

Training and assessment systems are interactive narrative systems intended for training people's skills and assessing their knowledge within a given field of expertise. Training and assessment systems often integrate several modalities and apply full-motion video to create real-life situations in which the user is to be trained or assessed. The idea is to confront the user with different situations where decisions or actions are needed. The hardware used for these systems may be desktop computers with ordinary input and output devices or complex and dedicated display systems.

A simple example is a system for training prospective automobile drivers. The objective could be to train people in evaluating certain traffic situations and making decisions when driving an automobile. Such a system might involve different modalities such as video-clips from drives, sounds from the engine and other vehicles, 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.

We have conducted two experimental evaluations that involve design of training and assessment systems. These experiments are described below.

2.2. Experimental evaluation of an object-oriented design method

Setting: A general object-oriented analysis and design method, cf. [19], was applied in a study of the design process of a training and assessment umpiring system. We will refer to this study as the Umpire System Project. The purpose of the experimental study was to evaluate the relevance of a typical software method when designing interactive narrative 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.

Application domain: 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 make calls almost immediately in a developing situation by applying a complicated set of rules. Within each situation, only one call is correct according to those rules. This implies that the user has to identify the correct call among a number of different options. The situations are designed and described to reflect typical and relevant umpiring situations. Roles of participating race boats and settings need to be described and defined in order to set up and shoot the video footage. 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.

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 [20].

2.3. *Experimental identification of ideas for improvement*

Setting: General narrative knowledge and story-telling techniques, cf. [2,21–23], were applied in the design of a training and assessment manager system (further explained in Section 4). We will refer to this study as the manager system project. The 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 how to approach interactive narratives systems design from another perspective, and thereby provide ideas for solving 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 an operational version of the system. The design process was started in January 1997 and ended in December 1997.

Application domain: 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 purpose of this system was to 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. For each situation, the designers need to define relevant actions and possible decisions that would be typical for a manager in the company. Hence, the testing of general skills imposes that no decision is more correct than other decisions. Decisions will on the other hand produce a profile reflecting personal skills of the user. At the same time, characters and settings need to be described and defined in order to shoot the video footage.

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 interactive narrative 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. Two other persons, including one of the authors of this article, conducted the interview 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-hours and it was fully transcribed into a document of 21 pages. This document is publicly available on the World Wide Web [24]. In addition, the design documentation was studied, and the design team answered further questions in a couple of short, informal sessions.

2.4. *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 [25]. The product may be a document, a revised version of a document, a prototype or some other outcome that can be identified. For each

phase it was described how it was conducted and what the product was.

In order to analyze the methodological problems that were encountered using the object-oriented modeling method, we developed a framework with six design elements. These elements were defined by means of the literature on object-oriented design and interactive narrative systems design.

Booch defines design as the disciplined approach one uses to invent a solution for some problem, and the result of design as the provision of a path from requirements to implementation [26]. Jacobson et al. describes a model as an abstraction of a system specifying the modeled system from a certain viewpoint and at a certain level of abstraction [27]. One viewpoint could be the design of the system. These general points imply that the design of a training and assessment system has to provide the developers with a good and adequate basis for programming the system. At this point in the development process, all major parts of the system should have been analyzed and described, and it should be clear how the content and the functionality of the system fulfill the requirements to the system. Based on this, we have identified the six key elements in the design of a training and assessment system that are illustrated in Fig. 1.

The results of this analysis are presented in the following section. In addition, we have used the frame-

work to summarize how a successful training and assessment system was designed. These results are presented in Section 4.

3. Experimental evaluation of an object-oriented design method

In the Umpire System Project, all activities were carried out exactly as prescribed by the object-oriented analysis and design method OOA&D, fully described in [19] and briefly summarized in [28]. This method combines the strengths of three widely known methods: the object-oriented concepts from [26], the relation between real world objects and the dynamic model inside the computer system from [29] and the practical guidelines for object-oriented analysis and design from [30].

The design process in the Umpire System Project consisted of four major phases: system definition, problem domain analysis, model component design, and prototype implementation. 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 computerized system is, used to administer, monitor, or control. Second, the application domain that is the organized group of users

Design element	Description
1. Problem domain model	The system models the objects that the user observes or acts upon
2. Functionality	The user interacts with the system by activating the functions that are available
3. User interface	The overall collection of interface elements, the navigation between them and the layout of each individual element
4. Media assets	The interface elements are composed from a collection of smaller elements that each involve one modality
5. Situation model	The system presents real-life situations in which the user will be immersed and act
6. Overall story	The situations are parts of a larger story with a certain order and causality between individual situations

Fig. 1. Design elements for training and assessment systems.

that apply the computerized system to support their work. Third, the computerized system that is a collection of software components that maintains a dynamic model of the problem domain in order to provide relevant information to the users in the application domain.

The purpose of the problem domain analysis phase was to specify requirements to the system’s model that represents the problem domain. In this phase, the designer started to identify and specify the key classes of the problem domain. He decided that the system should contain classes that model the user of the system and the calls (the name of a decision that is made by an umpire) made by this user. The user of a system is generally not part of the problem domain. But a typical requirement to a training and assessment system is that the user’s actions and decisions are registered. This is necessary in order to provide data for later evaluations of the user’s umpiring performance.

The designer also decided that the system should integrate classes that represented various match race situations. Here, a situation denoted a sequence of events that eventually leads to an incident where one or both boats protest against the other. This was a fundamental conceptual change, because initially the term situation was used to define a sequence of protests and each single protest from the boats was called a protest.

A match race is conducted on a standardized course that is established on the water where it is marked with floating buoys. Different match race situations occur on different parts of the course. The designer decided to define a class called course, which models a set of situations related to a specific part of a match race course. An object from the course class should aggregate a collection of situations that usually occur in that part of the course. The modeling activity resulted in the class diagram illustrated in Fig. 2(a). The notation in the diagram is consistent with UML [31]. The diagram facilitates overview over the problem domain but it is far

from giving any concrete impression of the working of the interactive narrative system.

The modeling of behavior for the classes caused some problems. The behavior for the situation class was actually quite simple since no user actions or inputs would cause any changes or updates in the classes. On the other hand, the designer found it difficult to distinguish between the behavior of classes modeling match race situations and the behavior of classes modeling the user’s usage of the system. Furthermore, objects of the situation class cannot be created, updated, or deleted by the user and hence the creation and deletion of these should not be modeled. As illustrated in Fig. 2(b), these objects were designed to wait in a passive state until they were actually displayed to the user in which case they became active.

The behavior of classes was described by means of state chart diagrams which is UML’s version of state transition diagrams [31]. An example of such a diagram is shown in Fig. 2(b). The state chart diagrams were useful for describing the specific behavior of objects but they provide no overview of the behavior of the complete system.

The problem domain analysis phase comprised nearly all of the analysis work that was conducted in the project. The typical object-oriented design process involves a comparable amount of effort spent on analyzing the application domain. The designer also started on that, but it quickly turned out that the use situation with the umpire system is very simple. With a typical computerized system, there will be a whole array of use situations that may be carried out by different actors. So the application domain is described by a collection of use-cases and an overall use-case diagram that relates different actors to the use-cases they are carrying out. But with the umpire system, there is only one single actor, which is the user that is working with the system. And the work with the system is simulating a specific and specialized task where the interaction on use-case level is that the system displays a situation to

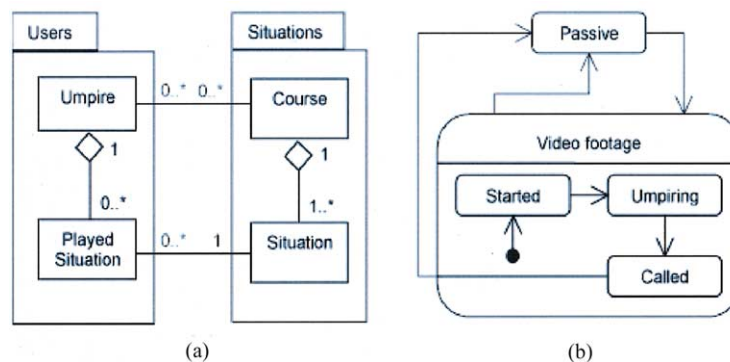


Fig. 2. (a) The class diagram for the umpire system. (b) The behavior of the situation class.

the user and the user then makes a call. Thus there is only a single use-case and that one is very simple. Therefore, the requirements related to the application domain were only outlined. This limited relevance of the use-case based description might be a general characteristic of training and assessment systems.

The model component 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 the 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 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 applications that are built with it. The prototype contains one match race situation that shows the maneuvering of two boats over a period of about 1 min and ends with a protest from one of the boats. This situation is made as a video clip, being recorded from the position that an umpire boat would typically be

in during that situation, combined with two soundtracks containing the voice of a fellow umpire and a wing observer. The prototype was used to decide that it is possible to implement the system, and that the design creates a relevant use situation.

The main experience from this experiment is summarized in Fig. 3.

4. Experimental identification of ideas for improvement

The Manager System Project was fundamentally explorative in its approach to system development. The purpose of the study of this project was to evaluate alternative approaches to the design of training and assessment systems. The key techniques employed were founded in narrative theory including story writing and movie making, cf. [2,23]. We denote this as general narrative knowledge since it combines basic ideas and techniques found in structuralism theory [21,22]. Structuralism theory distinguishes between the story of the narrative and the discourse of the narrative. The story defines the events and existents, e.g. actions and characters. In the project, the story would include descriptions of individual scenes, characters acting in these scenes, the appearance and characteristics of the

Design element	Strengths and weaknesses
1. Problem domain model	Good support of design of the model. Strong support for structuring and classification of the software components.
2. Functionality	Limited support as the application domain analysis gave only limited understanding of the system in use and few ideas to functions.
3. User interface	Systematic support for designing the user interface.
4. Media assets	No means for design or production of assets. In addition, it was difficult to determine the component of the system they should be placed in.
5. Situation model	Problems in identifying and describing situations. Too simple descriptions of situations because no user action will update or delete information related to situations.
6. Overall story	No means for design and production of the overall story in the system.

Fig. 3. Results of the experimental evaluation of an object-oriented method.

setting for the scenes etc. The discourse defines the means by which the story is communicated. In the project, the discourse exemplifies the different paths through the narration and the causality between the scenes of the story.

The project incorporated three major phases: prototyping, story creation, and implementation. 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”. In addition, it gave the developers a basic idea of how to approach an assignment of this character. The prototype was developed within a week by means of an author tool, and it can be characterized as an experimental prototype, cf. [32]; 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 training and assessment 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 descriptions and act descriptions. 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 setting, the duration of the scene etc. These descriptions would form the basis for later directing and shooting of video footage. In Fig. 4, an abstract illustration of the produced documents is illustrated. At the top right corner of the figure two screenshots illustrate a situation where the employees of a department are having a

meeting. The video footage of these scenes was produced upon the scene descriptions. In this situation, the scene descriptions would include information on the cast, six employees (two women and four men), how they are dressed etc., the setting, they are sitting in an office around a table etc. All things necessary would be described in the scenes. At the macro level, a script was created to describe the flow of the acts that tied the individual 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 Fig. 4, this is illustrated by textual descriptions of scenes, their implementation as video sequences, and the overall structure between these video clips and user decisions. The dark boxes and arrows represent a path followed by one user of the system (from scene A, the user has chosen a selection leading to scene B2 and so forth).

The tool enhancement phase involved the construction of an enhanced implementation platform. The purpose of this phase was to enable designers to modify the implementation of the story within the tool; there was a lot of generality in story creation with respect to integrate media elements and in the specification of how and when 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.

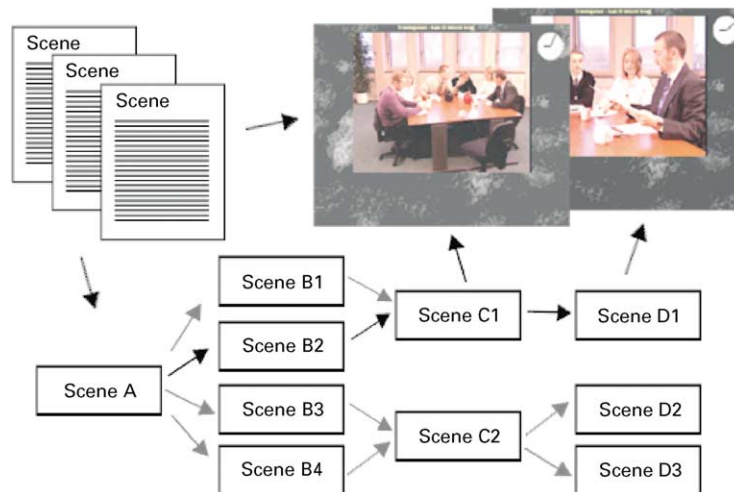


Fig. 4. The story creation documents for the Manager System Project.

Design element	Strengths and weaknesses
1. Problem domain model	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.
2. Functionality	No explicit support for the design of functional requirements.
3. User interface	No techniques or means provided which lead to arbitrary design of the user interface. However, the user interface was coherent and standardized.
4. Media assets	The shooting (and hence creation of assets) of scenes could be done in one go. Their documents provided a sufficient basis for producing the assets.
5. Situation model	The creation of scenes and acts made it possible for the designers to continuously evaluate the realism of the system in use.
6. Overall story	Good and solid support for the design and creation of the storytelling part of the system.

Fig. 5. Results of the experimental evaluation of a process based on general narrative knowledge.

As a drawback, since the technical system was not documented the design team was very dependent on individual programming skills. One of the programmers did most of the programming and the designers were aware of the potential risks associated with this fact. In addition, they had technical problems with the stability of the system where they every now and then experienced system breakdown. They were not immediately able to identify and locate the problem.

Summarized, it can be concluded that the application of general narrative knowledge for the design of a training and assessment system had the characteristics listed in Fig. 5.

5. Conclusion

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 interactive narrative systems design. The literature *does* include proposals for methods, cf. [11,12,33–35], 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 an interactive narrative 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 or the application domain. Finally, the production of the media 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.

The identified weaknesses of the object-oriented modeling method was the focus of a second study of another training and assessment system design process. This design process was conducted by applying 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 methodologically 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. 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.

A number of experimental properties limit the general validity of the above results. First, the two systems in the empirical studies share a number of similarities as interactive narrative systems but they also have some fundamental differences. Both systems involve the use of video footage depicting real life situations in which the user has to make decisions. However, the complexity of the situations in the manager system is higher than in the umpire system since no actions are predefined by the nature of the situation. In the umpire system, the designers applied the rules for match racing that determined a correct call for each situation. This was not the case for the manager system where the designers had to identify situations and related relevant actions. This makes the design and construction of the situations for the manager system more complex. Second, the approach taken in the manager system project was inspired by general narrative knowledge but not controlled by any method. The techniques and ideas applied were highly influenced by structuralism theory. However, since the design process was not supported or controlled by any well-defined method as in the case with the umpire system, it is difficult to separately identify the influence of the participating designers and to make absolute conclusions about their work and results.

The research has revealed some potential avenues for further work in this field. First, a combination of the two approaches may seem to be an obvious way for further improvement in the field of interactive narrative systems 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 [6]. Third, another study could deal with implications concerning the usability

and final acceptance of the systems. The narrative approach in the manager system project was favored because the designers were able to verify and test the realism in use early in the design process. In addition, the narrative approach enabled designers to produce assets for the system. However, how is this related to the final acceptance of system and does it influence the usability of the system?

Our study indicates that object-orientation can be useful to some extent in the design of interactive narrative systems. Object-orientation helps designers to organize and describe properties related to the purpose of the system, and description of problem and application domains. However, object-orientation faces some weaknesses in dealing with the aspects of creating stories including describing the story elements and the discourse.

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. 9600869. Steen Andersen and Peter Stegenborg Larsen did the implementation of the Umpire System prototype, and Ian Peter Semey, Gabriel Schoenau Hansen, and Claus Rosenstand developed the Manager System. We are very grateful for their contributions. We would also like to thank the anonymous reviewers of this article for valuable comments on earlier drafts. Finally, we would like to thank Lars Bo Eriksen for participation in earlier versions of this article.

References

- [1] Rollings A, Morris D. *Game Architecture and Design*. The Coriolis Group, 2000.
- [2] Galyean TA. *Narrative Guidance of Interactivity*. Ph.D. Thesis. Massachusetts Institute of Technology. Media Arts and Science Program, 1995.
- [3] Umaschi Bers M, Ackermann E, Cassell J, Donegan B, Gonzales-Heydrich J, DeMaso DR, Strohecker C, Lualdi S, Bromley D, Karlin J. *Interactive Storytelling environments: coping with cardiac illness at Boston's Children's Hospital*. Proceedings of the CHI 1998 Conference on Human factors in computing systems, 1998. p. 603–10.
- [4] Benford S, Bedreson BB, Åkesson K-P, Bayon V, Druin A, Hansson P, Hourcade JP, Ingram R, Neale H, O'Malley C, Simsarian KT, Stanton D, Sundblad Y, Gustav T. *Designing Storytelling Technologies to Encourage Collaboration Between Young Children*. Proceedings of the CHI 2000 Conference on Human factors in computing systems, 2000. p. 556–63.

- [5] Skov MB, Eriksen LB. Evaluating software engineering modeling concepts for interactive narratives design. Behind the scenes of multimedia production: methodologies of virtual inhabited 3D worlds. London: Springer-Verlag (forthcoming).
- [6] Pauen P, Voss J, Six H-W. Modeling hypermedia applications with HyDev. In: Sutcliffe A et al., editors. *Designing effective and usable multimedia systems*. Boston: Kluwer, 1998. p. 23–39.
- [7] Russo NL. Expanding the horizons of information systems development. In: Baskerville et al., editors. *Organizational and social perspectives on information technology*. Dordrecht: Kluwer Academic Publishers, 2000. p. 103–12.
- [8] Skov MB, Stage J. Using software engineering approaches to model dynamics in interactive software systems. *Virtual interaction: interaction in virtual inhabited 3D world*. London: Springer, 2001. p. 404–21.
- [9] Wilson M. Multimedia design: from tools for skilled designers to intelligent multimedia design systems. *Proceedings of the IFIP 13.2 Working Conference—Designing Effective and Usable Multimedia Systems*, 1998. p. 1–4.
- [10] Webb BR. The role of users in interactive systems design: when computers are theatre, do we want the audience to write the script. *Behaviour and Information Technology* 1996;15(2):76–83.
- [11] Sutcliffe AG, Faraday P. Designing presentation in multimedia inter-faces. In: Adelson B, Dumais, Olson J, editors. *Proceedings of Computer-Human Interaction Conference '94*, 1994. p. 92–8.
- [12] Sutcliffe AG, Faraday P. Designing effective multimedia presentations. In: Ware C, Wixon D, editors. *Proceedings of Computer-Human Interaction Conference '97*, 1997.
- [13] Fairley RE. *Software engineering concepts*. New York: McGraw Hill, 1985.
- [14] Pressman RS. *Software engineering: a practitioners approach*. New York: McGraw Hill, 1996.
- [15] Sommerville I. *Software engineering*, 4th ed. Workingham: Addison-Wesley, 1992.
- [16] Sutcliffe A, Ziegler J, Johnson P. *Designing effective and usable multimedia systems*. Dordrecht: Kluwer Academic Publishers, 1998.
- [17] Sutcliffe AG. User-centred design for multimedia applications. *Proceedings of International Conference on Multimedia Computing and Systems (ICMCS'98)*, vol. 1. IEEE Publications, 1998. p. 116–23.
- [18] Basili V. The role of experimentation in software engineering: past, current, and future. *Proceedings of the 18th International Conference on Software Engineering*. Los Alamitos, California: IEEE Computer Society Press, 1996. p. 442–9.
- [19] Mathiassen L, Munk-Madsen A, Nielsen PA, Stage J. *Object-oriented analysis and design*. Aalborg, Marko, 2000.
- [20] Eriksen LB, Skov M, Stage J. *Object-oriented multimedia system development: Documentation*, 1998. Available through the WWW at the following URL: <http://www.cs.auc.dk/~jans/matchrace/>.
- [21] Chatman S. *Story and discourse—narrative structure in fiction and film*. Ithaca and London: Cornell University Press, 1993.
- [22] Genette G. *Narrative discourse*. New York: Cornell University Press, 1988.
- [23] Laurel B. *Computers as theatre*. Reading, MA: Addison-Wesley, 1993.
- [24] Eriksen LB, Skov M, Stage J. *A Multimedia System Development Project: Documentation*, 1998. Available through the WWW at the following URL: <http://www.cs.auc.dk/~dubois/manager/>.
- [25] Andersen NE, Kensing F, Lassen M, Lundin J, Mathiassen L, Munk-Madsen A, Sørgaard P. *Professional systems development*. New Jersey: Prentice-Hall, 1990.
- [26] Booch G. *Object-oriented analysis, design with applications*. 2nd ed. Menlo Park, CA: Benjamin/Cummings, 1994.
- [27] Jacobson I, Booch G, Rumbaugh J. *The unified software development process*. Reading, MA: Addison-Wesley Longman, Inc., 1999.
- [28] Mathiassen L, Munk-Madsen A, Nielsen PA, Stage J. Modelling events in object-oriented analysis. In: Patel D et al., editors. *Object-oriented information systems*. Berlin: Springer, 1995. p. 88–104.
- [29] Jackson M. *System development*. New Jersey: Prentice-Hall, 1983.
- [30] Rumbaugh J, Blaha M, Premerlani W, Eddy S, Lorenzen W. *Object-oriented modelling and design*. New Jersey: Prentice-Hall, 1991.
- [31] Booch G, Rumbaugh J, Jacobson I. *The unified modeling language user guide*. Reading, MA: Addison-Wesley Longman Inc., 1999.
- [32] Floyd C. A Systematic Look at Prototyping. In: Budde R, Kuhlenskamp K, Mathiassen L, Züllighoven H, editors. *Approaches to Prototyping*. Berlin: Springer, 1984.
- [33] Donaldson J, Cowderoy A. Towards multimedia systems quality. In: van Veenendaal E, McMullan J, editors. *Achieving software product quality*. 1997. p. 189–205.
- [34] England E, Finney A. *Managing multimedia, project management for interactive media*, 2nd ed. Harlow: Addison-Wesley, 1999.
- [35] Hemsley J. Multimedia system quality problems, opportunities. In: van Veenendaal E, McMullan J, editors. *Achieving software product quality*. Den Bosch, 1997. p. 179–88.