Case Study based evaluation of Hypermedia Design Methods

Phyo Kyaw
Centre for Software Maintenance
Department of Computer Science
University of Durham
Tel : (0191) 374 2549
E-mail : *Phyo.Kyaw@dur.ac.uk*

Cornelia Boldyreff
Centre for Software Maintenance
Department of Computer Science
University of Durham
Tel: (0191) 374 2538
E-Mail : *Conelia.Boldyreff@dur.ac.uk*

Abstract

This paper evaluates different hypermedia design methods by redesigning an existing web site (reverse engineering) to a new structured web site and by applying the methods to create a new web site (forward engineering). It also illustrates how complex and large existing web sites can be redesigned into maintainable structures using systematic design methods. Since most of the web sites are already implemented, it shall be a simple presentation for who intends to create or update their WWW sites and takes full advantage of structured design methods. The paper presents two hypermedia design methods, namely RMM and OOHDM, including their overview and their detailed design stages to reverse engineer an existing design. Since this paper has only focused on design issues, any implementation details will be omitted.

1. Introduction

In recent years, the use of WWW (World Wide Web) and hypermedia applications become distinctively increased. Many organisations, institutes and others have implemented a large number of web sites and they are still producing many web sites for different aims and objectives. However many WWW designers and novice users are currently developing their web applications with their own knowledge without applying any formal design methods. Therefore potentially the design of large applications become remarkably complex and moving towards maintenance problems. Later in this paper we will be showing how existing web sites can be redesign using structured methods by applying to an existing web site as a case study.

2. Case study background

The Software Engineering Group (SEG) Website is designed to present information about the software development projects. These projects are given to the students to obtain experience of a software production environment similar to that found in the computer industry, i.e. working as a part of a design and implementation team to tight schedules. The SEG Website includes the following information.

- Project Structure – Project structure includes information concerning project aim, attributes of the project and group organisation.
- Project Outline – It presents information about the system that the department wished to develop.
- Group allocation – It includes information about students and supervisor group allocation details who are developing the project.
- Project Phases – Project phases are presented and deliverable to be done. It also includes information about the development time (in weeks) and each project development stages to be completed.
- Intermediary Tasks – This section presents the group-work tasks to be carried out throughout the year. It also includes information about each task to be completed in each week.
- File structure - All documents produced will be stored (and delivered) in BSCW workspaces. Each member of a group will be allocated a BSCW UserID and password for use in a private workspace. All work for this project should be undertaken under this UserID and should have the fixed project file structure. All documents for submission should be produced in HTML, and when they are ready for submission they should be placed in the Documents folder within the groups' private workspace.
- SEG Lab – To carry out various development tasks, students need to use SEG laboratories. This section presents detail information about booking information for booking of SEG Labs.

3. Reversing engineering using RMM

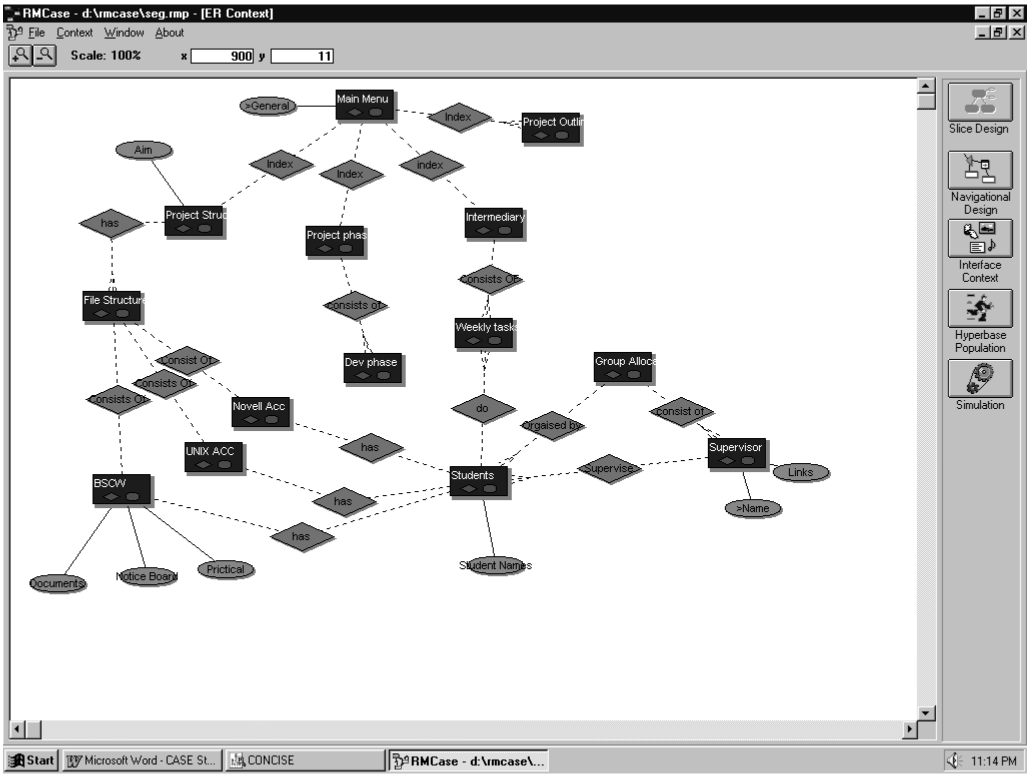
The reverse engineering of the SEG Web page was started by the analysis of application domain. The analysis mainly includes current Web page design, information grouping and navigational structures. It also consists of the identification of domain objects and their navigational links. Furthermore, the analysis is made on the current use of navigational access structures such as indices and guided tours. However it takes considerable amount of time to evaluate every detail sections of the site. Therefore the reverse engineering was made on top level view of the current design. The RMM is suitable for applications with hypermedia front-ends of database or legacy applications. Hence SEG Website is not welled suited to apply RMM method since the data remains unchanged over time. However the main idea of this chapter is to evaluate all the different design and construction phases using various hypermedia design methods. Consequently, the new design of SEG page can obtain benefits from RMM design guidelines.

Once the analysis of the problem domain was completed, the reverse design was started by reversing the current web site into Relational Management Data Model (RMDM).This process includes the following stages.

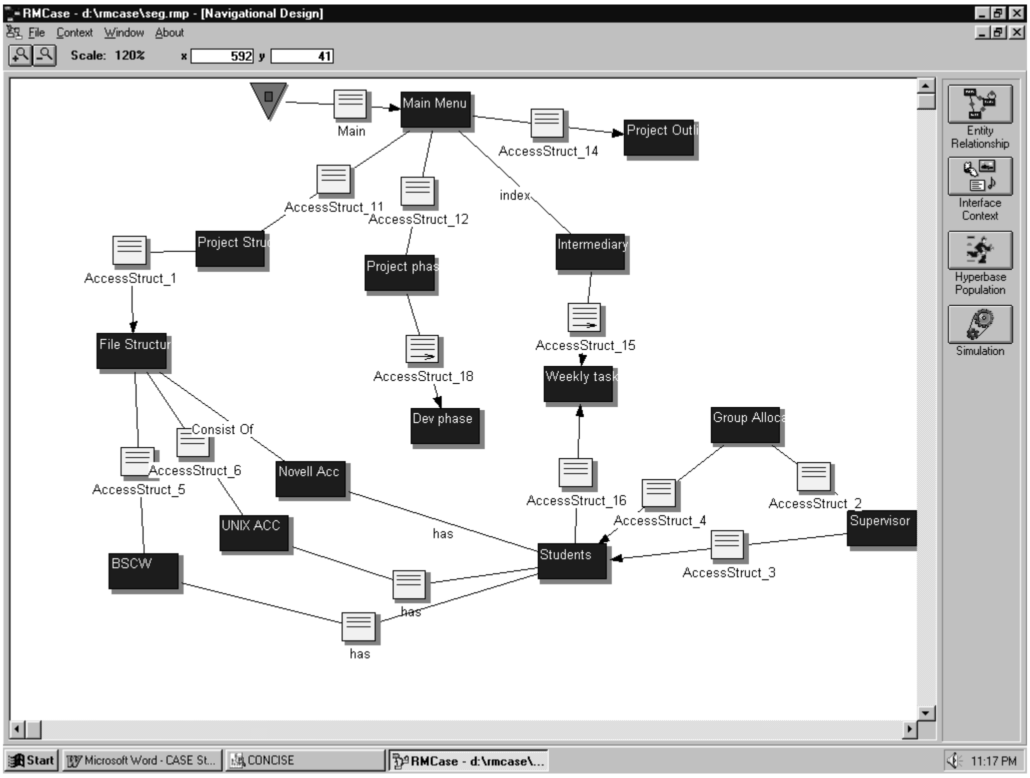
1. Each section is grouped based on RMDM modelling primitives. As describe in previous chapter, RMDM modelling primitives consists of E-R domain primitives, more detail slices and associative

relationships. (see figure 1)
2. The second step is to constructs ER and slices based on current Web side. This includes grouping of information objects and attributes, designing of navigational access structures and relationships (see figure 2).
3. After the current design has been produced, the design is transformed into new structured design using RMCase, which is a CASE tool for RMM design method. See figure (3). This includes the creation of new ER designs, slices and navigational links with the assistance of RMCase.
4. Once the basic informational structure and navigational design has completed, the new design is tested and evaluated.

3.1 New ER design, constructed using RMCase



3.2 The development of navigational design



3.3 Evaluation of resigning the SEG Web site using RMM

Since RMM is aim for the development of systems that interact with databases and integrated with other various information systems, SEG was not well suited with RMM design methodology. RMM is only suited the hypermedia development of large and dynamic database information systems. SEG Web site was developed without any formal hypermedia design approaches and there are redundant data and very little use of hypermedia navigation. However during the redesign stage, using RMM design method, many bugs and inconsistencies were eliminated in terms of grouping data and navigation. The use of RMM Slices helps the design to split information entities into meaningful group of attributes. This eliminates the disoriented information, which is undesirable for users. The use of RM-CASE, that supports RMM design and development, also assists the redesign of SEG Web. Since RM-CASE tool is still a beta and incomplete version, no user-interface design can be constructed. The use of General tools such as HTML editors can offer some support for interface design implementation.

4 Reverse engineering using OOHDM

In this section, the SEG Web-Site is reverse engineered and redesigned using OOHDM. For the purpose of evaluation of the usefulness and limitations of OOHDM. The evaluation includes: the object abstraction capabilities, reusability of objects, portability, notations, navigational perspectives, interface design constructions etc.

Furthermore the nature of the hypermedia fields, design problems of SEG Website are presented in previous section. Therefore, this section will not elaborate the requirements of the CASE STUDY nor actual implementation artifacts but stressing the design steps, development cycles, navigational views of the SEG Website and the evaluation of OOHDM.

Brief outlines of development stages are listed bellow to obtain clear orientation of OOHDM design process. However the full description of OOHDM is described above.

- The first design step of OOHDM is the development of conceptual model using object oriented primitives such as classes, subclasses, methods and attributes.
- Once the conceptual model is completed, the navigational model is constructed using both features from object oriented method and hypermedia design methods. It includes:
  1• The construction of navigational node classes with different perspectives, which may include one or more attributes from all or part of the conceptual classes.
  2• Different navigational views are defined for one conceptual schema to allow the different users to perform different tasks.
  3• The construction of various types of links including application links, structured links and access structures.
- Once the navigational structure has been constructed, abstract interface design is defined to make perceptible to the users. It includes interface objects, ADVs and other multimedia objects.
- The products from abstract interface design can be implemented using any Web page authoring tools.

Reverse engineering of SEG Website
The main steps of reverse engineering are to reverse current implementation to design and rebuild that design to new implementation. Since the current implementation is reversed to design in previous method (see RMM), the same design schema will be used to build new design and new implementation.

4.1 New Conceptual design

During Conceptual design a model of SEG Web site is designed using objects, classes and relationships between class. Objects are the instances of classes, and sub-classes are the types of classes.

As shown in figure [4.1], boxes represent the classes, links represent relationships and small diamonds indicate aggregation. The small classes project 'phase' and 'tasks' are part of the project schema. The main 'project' class is part of the schema of the SEG Web site. In this stage the representation is only to capture the domain semantics without any navigational structures for users tasks. Therefore there are no node classes and links are shown in this stage. Figure [4.2] shows the simplified version of conceptual design of the SEG Web site. It consists of three main classes, 'project', 'person', and 'file system'. As shown in the figure, 'student' and 'supervisor' are types of 'person' class and 'file structure' class consists of three subclass, which inherit the main class. Each class consists of various attributes that can be implicitly defined in other classes as a relationship. Each attribute can have more than one perceptive to be presented to the users. For example, 'project.structure' can be presented using both text and diagrams. The relationships between objects can be mapped with a navigational view. However the navigational classes require more link structures to present information from various perspectives. For instance, there are no entry nodes for either students or the projects.

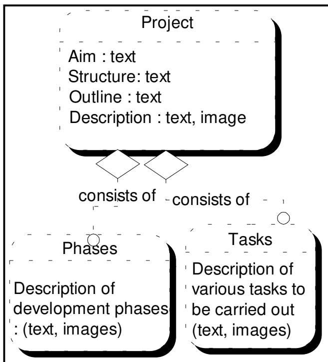


Figure 4.1:Conceptual Schema of Project (part of SEG project)

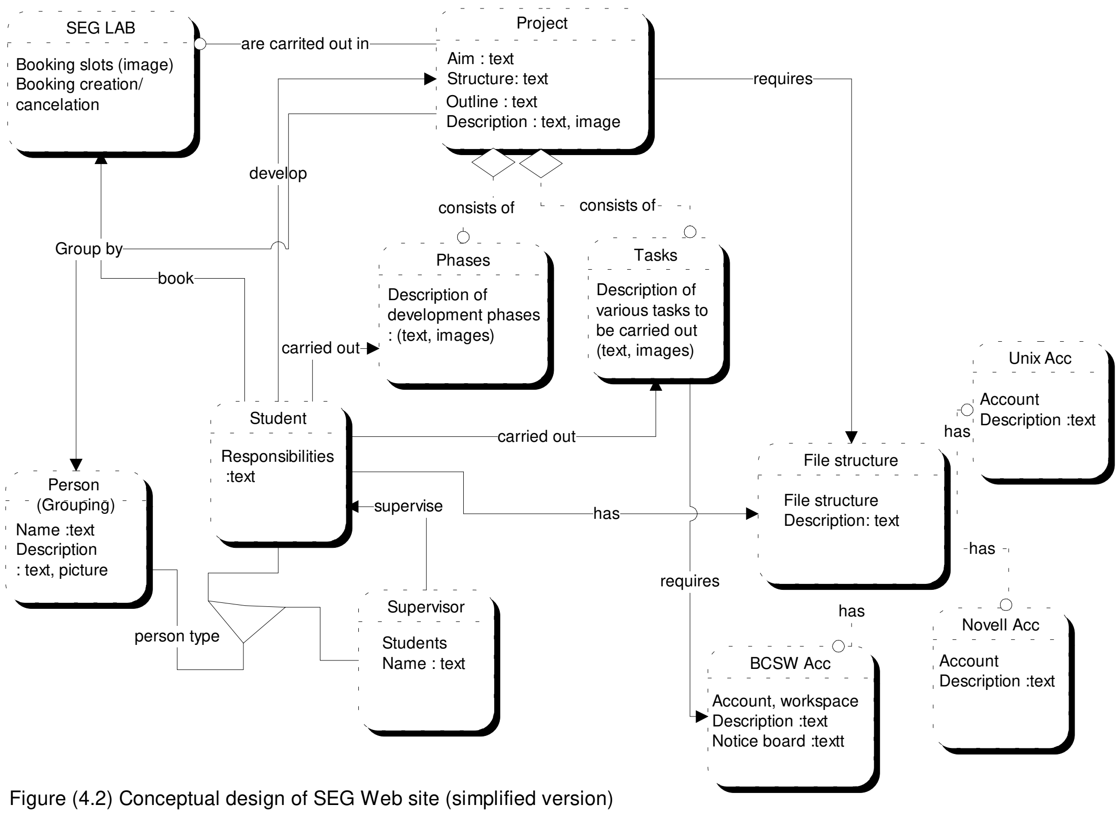


Figure (4.2) Conceptual design of SEG Web site (simplified version)

During the conceptual design the construction of aggregation and inheritance is vital to simplify complex application domain. However the development of conceptual design requires many skills and an object orientation background. Therefore when conducting the design process, the designer requires adequate formalisms and tools for handling the formal aspects of the design with the support of designer’s incremental and feedback loops (Nanard and Nanard 1995). The adequate design environments are essential for the productivity of the designer in each design step.

4.2 New Navigational Design

During this step, navigational node class and links are constructed based on user tasks. The navigational classes can be derived from one or more conceptual classes from previous stage. When designing navigational models, different navigational views (models) can be built for the same conceptual schema to present to different users(Schwabe, Rossi et al. 1996). However for the SEG new design, two navigational views (models) are constructed for the same conceptual schema. They are 'students' perspective and 'visitors' perspectives. Defining a navigational model consists of various classes such as navigational nodes, links, access structures, etc. Navigational nodes are windows for the users that may contain various attributes and anchors. This means that a node may consists of attributes from one or more conceptual class. Links represent the relationship between nodes. Link classes can be defined by specifying link attributes and behavior, source and target nodes and cardinality.

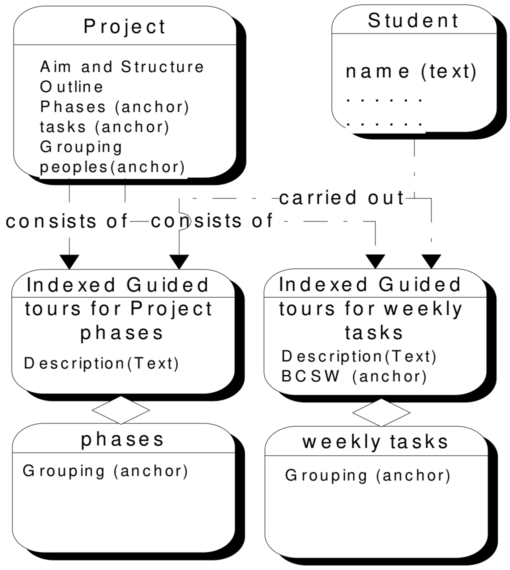


Figure [4.3a] :Navigational Schema of Project (part of SEG project)

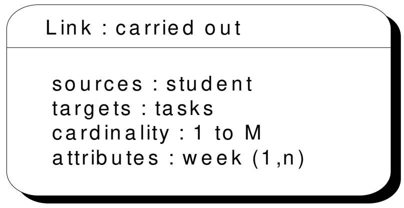


Figure [4.3b] link class

Figure (4.3a) shows nodes and links that are part of the navigational design of the SEG Website. Boxes represent navigational node classes and the doted links represent the links between nodes. Nodes are derived mainly from corresponding conceptual classes. Figure (4.4) represents how navigational nodes are derived from conceptual classes. In the figure, each node consists of one or more attributes (components and units) from one or more attributes. For example navigational node ‘project’ is derived from ‘project’ conceptual class. Node class ‘project’ contains attributes such as project aim and structure and anchors to other classes. Since project ‘aim’ and ‘structure’ information will not be frequency seen by the students, they are embedded in the ‘project’ node. However for ‘visitors’ view project ‘aim’ and ‘structures’ are presented explicitly as aggregate to ‘project’ node. The attributes and anchors of a node class can be any media types (text, graphics, sound, etc). However the detail apparent of each attribute can be designed in user interface stage. The attributes of each navigational node can be derived from single conceptual class or combination of different conceptual classes.

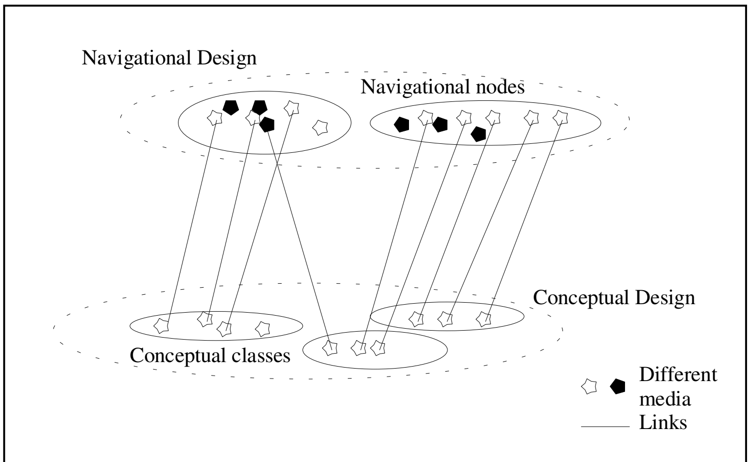


Figure (4.4) Conceptual design to Navigational design

Nodes are interrelated by application links class and access structures. The rationale for creation of links and access structures are to explore fully the power of hypermedia and to stimulate the user’s interest by implementing various types of connections among the nodes (Schwabe, Rossi et al. 1996). The links can be created for various reasons. These includes the representation of relations between object, associations, annotations and sequential links between objects (Signore 1996). In figure (4.3a) relationship between 'student' node and 'task' node is presented by 'carried out' link class. As shown in figure (4.3b) the 'carried out' class consists of source: student, target: tasks, cardinality: 1 to N and attributes: weeks(1-n).

In addition to link classes, figure (4.4) shows access structures. Access structures can be indices, guided tours or indexed guided tours. Access structures help the reader to access the information easily. For example, project phases are presented using indexed guided tours. Access structures can be added freely by the author based on different perspective. For instance, in figure (4.4), additional access structures are provided direct access to require information. Therefore students can directly access to detail tasks, project phases and BCSW folder structure. However, for 'visitors' point of view, project aims and structures are more useful than detailed weekly tasks.

The design that was constructed for this CASE STYDY requires only static navigational structures. However the dynamic structure requires to define the navigational transformations that occur while traversing links(Schwabe and Rossi 1995). In OOHDM, D. Schwabe presented the use of Navigation Charts to construct dynamic navigational behavior(Schwabe, Rossi et al. 1996).

4.3 User Interface Design and Implementation

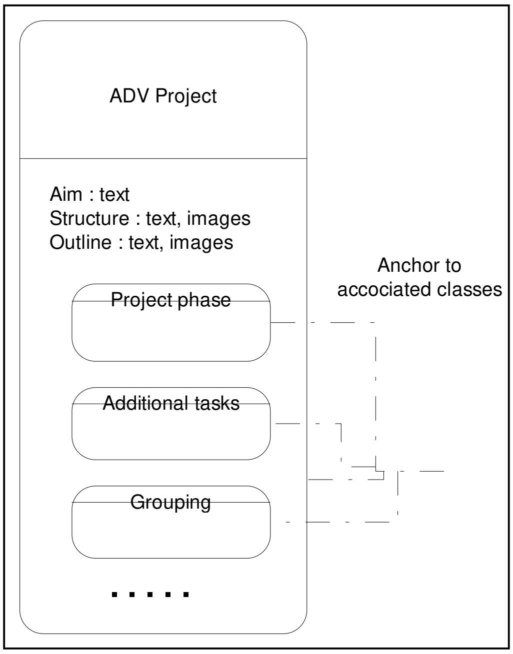


Figure (4.5) Simplify version of Abstract Data View (ADV) for 'Project' node

During this step, navigational classes are presented to the user by designing user interface. In this step, the new navigational design have to transform in a detail level to each interface metaphor and interaction styles. When designing interface objects, the author has to ensure what and how each object will be presented to the user. For example, what primitive (text, image, animated image or buttons, etc) will be used to provide the interface for a particular object. The consideration also includes the group of objects and how they are mapped to the navigational objects. Figure (4.5) shows general Abstract Data View (ADV) for 'project' node. It includes the definition of group links and perceivable objects that describe the aim and structure of the project. These objects may be presented by text, images or diagrams and may include embedded links. The use of ADVs assist the design and transformation of navigational objects to perceptible interface objects. It also helps to design interface objects independently from navigational objects. Therefore different interface aspects can be derived from the same navigational view.

4.4 Evaluation of OOHDM

During the development of SEG Web-site, OOHDM guidelines help to eliminate inconsistencies and mistakes. If the application is relatively straightforward manner in order to convert linear text into hypertext structures, OOHDM is not appropriate. OOHDM applies hypermedia features and design guidelines from HDM model with notations and principles from object-oriented design methods. The different between OOHDM and HDM is the use of object-oriented nature. However it adds the nature of object oriented techniques from object oriented approach. Therefore it has the power of object oriented concepts including abstract classes, methods, inheritance, aggregation, generalisation, etc. Moreover for the construction of interface design, it uses powerful abstract interface model (ADV charts) which allows the designer to build different interfaces for same navigational design. However there is no CASE tool available for the development of both interface design and translation of models to actual implementations. The one of the main advantages of OOHDM is that different navigational models can be derived from the same conceptual domain design, which can potentially add different perspective for different user views. However, generally speaking, since the detail design on the OOHDM considered to be complex, the development of the hypermedia system lies on the background skill and knowledge of the author and the design environment (Nanard and Nanard 1995). This means the author should not only be able to organise and structure different object-oriented models, but also should be able to observe and evaluate hypermedia structures in each design steps. RMM, which discussed in previous section, differs from OOHDM in that RMM is mainly based on hypermedia database applications where OOHDM can be applied for both retrieval-oriented information systems and other hypermedia systems. OOHDM also provides the development of interface objects that can be visualised in various ways, where RMM concentrates on navigational features.

5 Conclusions

The main contributions of our work includes to assist the designers and developers to be able to choose the write method and tools for any specific applications. We have described and tested major hypermedia design models and explore their benefits and limitations. We also described the features of hypermedia which can then apply for the development of WWW applications. We also describe how hypermedia application and models are related and shows their advantages and limitations including different notations and other evaluation criteria.

In this paper we have reverse engineered the currently developed (SEG) web site with hypermedia design models and evaluate the improvements in terms of quality and richness of functions. We also look at ideas of different navigational views for different collection of users and talks for the same conceptual design. Currently, WWW sites and applications are implemented without any formal design approaches and constructed mainly in an ad hoc approach. Our research is leading to designing new models and analysing existing hypermedia design model based on their usability criteria. In the future, we will be looking at different new approaches and models to improve the quality of WWW applications.

References

Balasubramanian, V. (1994). State of the Art Review on Hypermedia Issues And Applications.

Bichler, M. and Nusser, S. (1996). Modular Design of Complex Web-Applications with SHDT.

Bieber, M. and Vitali, F. (1997). "Toward Support for Hypermedia on the World Wide Web." Computer IEEE **30**(1): 62-70.

Brown, H., Ed. (1991). Hypermedia/Hypertext And Object-Oriented Database. London, CHAPMAN & HALL.

Bush, V. (1945). "As We May Think." The Allantic Monthly(July 1945).

Cowan, D. D. and Lucena, C. J. P. (1995). "Abstract Data Views, An Interface Specification Concept to Enhance Design for Reuse." IEEE Transactions on Software Engineering **21**(3).

Díaz, A. and Isakowitz, T. (1995). "RMCASE: Computer-Aided Support for Hypermedia Design and Development." International Workshop on Hypermedia Design.

Garzotto, F., Mainetti, L., et al. (1996). Information Reuse in Hypermedia Applications. The Seventh ACM Conference on Hypertext, Washington D. C.,VOL , ACM.

Garzotto, F., Paolini, P., et al. (1993). "HDM - A Model-Based Approach to Hypertext Application Design." ACM Transactions on Information Systems **11**(1): 1-26.

Hendrikx, K., Olivié, H., et al. (1997). Object-Oriented Hypertext Design: Authoring for Reuse. The 30th Annual Hawaii International Conference on System Sciences,VOL 3, IEEE.

Isakowitz, T., Kamis, A., et al. (1997). Extending the capabilities of RMM: Russian Dools and Hypertext. The 30th Annual Hawaii International Conference on System Sciences,VOL 6, IEEE.

Isakowitz, T., Strohr, E. A., et al. (1995). "RMM: A Methodology for Structured Hypermedia Design." Communications of the ACM **38**(8): 34-43.

Lange, D. B. (1994). An Object-Oriented Design Method for Hypermedia Information Systems. The 27th Annual Hawaii International Conference on System Sciences,VOL 3, IEEE.

McFadden, F. R. and Hoffer, J. A. (1994). Modern Database Management. Canada, The Benjamin/Cummings Publishing Company, INC.

Morris, M. E. S. and Hinrichs, R. J. (1996). Web Page Design: A Different Multimedia. USA, SunSoft Press.

Nanard, J. and Nanard, M. (1995). "Hypertext Design Environments and The Hypertext Design Process." Communications of the ACM **38**(8): 49-56.

Rossi, G., Schwabe, D., et al. (1995). An Object-Oriented Model for Designing the Human-Computer Interface of Hypermedia Applications. International Workshop of Hypermedia Design,VOL .

Rumbaugh, J., Blaha, M., et al. (1991). Object Oriented Modeling and Design, Prentice Hall Inc.

Schwabe, D. (1995). "The Object-Oriented Hypermedia Design Model." Communications of the ACM **38**(8): 45-46.

Schwabe, D. and Rossi, G. (1995). Building Hypermedia Applications as Navigational Views of Information Models. 28th Annual Hawaii International Conference on System Sciences, Mavi,VOL 2, IEEE.

Schwabe, D., Rossi, G., et al. (1995). Abstraction, Composition and Lay-out Definition Mechanisms on OOHDM. ACM Workshop on Effective Abstractions in Multimedia, San Francisco, California,VOL , ACM.

Schwabe, D., Rossi, G., et al. (1996). Systematic Hypermedia Application Design with OOHDM. Seventh ACM Conference on Hypertext, Washington DC,VOL , ACM.

Signore, O. (1996). Exploiting Navigation Capabilities in Hypertext/Hypermedia. The 29th Annual Hawaii International Conference on System Sciences,VOL 3, IEEE.

Takahashi, K. (1997). Analysis and Design of Web-based Information systems. Sixth International World Wide Web Conference,VOL , HyperNews.