A Tool for Designing Multimedia Systems.

Conference Paper · January 2001

Source: DBLP

CITATIONS
0

READS
179

4 authors, including:

Narciso Cerpa
Universidad de Talca
102 PUBLICATIONS **1,166** CITATIONS

SEE PROFILE

A Tool for Designing Multimedia Systems

Benjamin Pak, Sunny Tsang, Narciso Cerpa, and Rodger Jamieson

School of Information Systems, Technology and Management,
University of New South Wales, Sydney 2052, Australia

ABSTRACT

This paper aims to present the background and development of a software tool to assist the development of complex multimedia systems. After introducing the importance of multimedia design, the paper describes the four common phases of multimedia methodologies (Conceptual design, Navigational design, Abstract Interface design and Implementation), and compares and contrasts a selection of multimedia methodologies. This comparison and discussion assists the formulation of design criteria for the development of the Multimedia System Design Tool (MSDT) developed in JAVA by the authors. Finally the paper reports the validation of the MSDT software prototype using the development of a photo gallery as the evaluation design example.

Keywords: Multimedia, Methodology, Conceptual Design, Navigational Design, Abstract Interface Design, Software Tool.

1. INTRODUCTION

The World Wide Web has been one of the greatest developments in the computing and communications fields. Human beings are becoming more and more dependent on computers and the Internet. People are changing their whole concept of living, from home to work to their entertainment, such as human beings communicating with others using email and/or chat facilities [1]. Traditional media used in supporting interaction between humans and computers like sound, graphics and movies are now part of interactive multimedia which brings a new kind of energy to human society and information industries. Most companies have built web sites which promote their businesses, and there are also lots of e-commerce companies beginning to supply online services through the internet which will be one of the most important multimedia environment in this world. Therefore demand for quality multimedia systems is growing everyday. Multimedia has strategic importance in the convergence of computing, publishing, telecommunications, TV/Film industry, consumer audio/video electronics, and the emergence of 'New Media'. Interactive multimedia systems have become increasingly popular in different aspects of our society, whether it be personal or commercial. One sector revolutionized by the use of multimedia is education, evidenced by the now numerous online courses, interactive books (CD-Roms) and virtual machines for training skills. These systems can deliver information, news and instructions in a more efficient way.

2. MULTIMEDIA SYSTEM DESIGN

Designers face many problems during the process of designing multimedia systems. There are many systematic ways of solving and assisting in the development of multimedia applications. A good multimedia design methodology should assist the designer produce a multimedia system that helps users to organize and relate the information they need. The design of multimedia systems involves capturing and organizing the structure of large amounts of diverse information and making it clear and accessible to users. A multimedia design methodology must first focus on how the information should be connected, represented and presented. Secondly, it must consider the interaction of the users with the system, that is, the user interface [2,3]. When using a design methodology, each design activity is based on different concerns at each development phase and at the appropriate level of abstraction. Any design decisions should be recorded and traced back and forth in the development process. Most of the multimedia design methodologies have similar design phases, although the name of the design phases may be different in different design methodologies. Normally, most of the design methodologies have four common phases, they are Conceptual Design, Navigational Design, Abstract Interface Design and Implementation phases.

The conceptual design phase is responsible for organizing the information. The navigational design phase is mainly used for designing the navigation in the system. The abstract interface design phase contains both the user interface and the general interface design. Finally, the implementation phase is used for constructing an implementation of the design. Although the number and name of phases in different multimedia design methodologies are not the same, the ideas are similar except for some different techniques and the fact that some design methodologies are better able to support particular design phases than others. The following sections will describe how different multimedia design methodologies support these phases.

Conceptual Design Phase

Multimedia system design methodologies may be grouped into three categories: Object Modelling Techniques (OMT), Entity-Relationships (E-R) modelling, and others. A conceptual schema is used as a communication language between designers, implementers and users. Those multimedia methodologies using an approach based on OMT, may rely on graphical [4-6] and non-graphical semantics [7].

The Object-Oriented Hypermedia Design Method (OOHDM) is a model with graphical semantics for building hypermedia applications. At the conceptual design phase is intended to capture the domain semantics, but there is no need to pay too much attention to the types of users and tasks. The result of this phase is a class diagram specifying the participating classes and their relationships [4].

The Object-Oriented Design Method for Hypermedia Information Systems (Lange's methodology) design phase [5], which is called the 'Class Framework', consists of identifying the classes and refining them. The first step in identifying the classes is identifying their relevance from the problem domain. Brainstorming is used at this stage and every class that seems to be useful in designing this multimedia system should be written down. After a list of classes has been created, these classes are

classified into different types, such as irrelevant classes, unspecific classes, and redundant classes. The classes' relationships, attributes and operations are also identified.

The conceptual design phase of the Scenario-Based Object-Oriented Hypermedia Design Methodology (SOHDM) consists of Domain Analysis and Object Modeling [6]. In Domain Analysis, the context diagram (zero level DFD) is used to represent the system scope, and the Scenario Activity Charts (SAC) are used to identify hypermedia application requirements. Using Object Modeling, scenarios are converted into objects in the form of Class Responsibility Collaboration (CRC) cards.

Adams' hypertext instructional design methodology provides a "requirements stage" that allows the conceptualization of a multimedia system. This stage is for designers to focus on the key concepts, which are going to be presented by the multimedia system [8]. The material should be divided into nodes according to the key concepts and produce a graph of the overall structure of the system. A systematic approach for decomposing subject material into conceptual components, the object-oriented text decomposition (OOTD) is used to facilitate this process.

A combined approach for designing hypermedia systems makes use of formal notation and semi-graphical modelling [7]. This methodology uses process algebra for the mapping of actions, events and sequences. The process algebra must be logical, follow the rules of mathematics, and proofs are performed to ensure validity and correctnes. Once complete and validated, the mathematical formulae are translated into graphical notation to facilitate reading and understanding

According to HDM, an application domain is composed of 'entities' which are the components. Entities belong to a 'type', and they can be connected to other entities or components by 'links'. Links have three different types: structural, application and perspectives. Different types of links have different functionality. The structural link shows the hierarchical structure of entities. The application link reflects application domain relations by connecting entities or components to other entities or components. The perspective link breaks up entities into units. Units contain a reference context to information [9].

The Dexter Hypertext Reference Model describes a multimedia system as a hierarchy of nodes interconnected by links. These nodes may be implemented as cards in HyperCard, documents in Augment or articles in Hypermedia [10]. These nodes or data components have more specific and detailed content than the entities used in E-R diagrams. For example, each component has a "Presentation Specification" which clearly describes the type of media which is used to present this atomic item, and the details of this media like the scale factor for graphics or the volume of sound.

The Amsterdam Hypermedia Model (AHM) extends the Dexter model by adding to it the notions of time, high-level presentation attributes and link content. For example, the time duration for graphics or text is unlimited but for sound or movie is the length of playing time. Conceptual components can be divided into atomic or composite. An atomic component contains the information that refers to a data block (i.e. entity). A composite component specifies the information for a set of atomic or composite data block, and all data in the composite components' definitions must refer to the specified atomic components. Composite components have two presentation options: parallel composite which present all of their components parts simultaneously, and choice composite which will only present one of the components at one time [11].

Navigational Design Phase

Since the multimedia application must be used by a set of intended users who are trying to accomplish a certain set of tasks, the designer must reorganize the information represented in the conceptual model into a navigational model [4]. In OOHDM, the navigational design is expressed by using two schemas, the Navigation Context Schema and the Navigational Class Schema. Navigational context is a set of links, context classes, nodes and other nested navigational contexts. In the Navigational Class Schema, the navigable objects of the multimedia system are defined. The classes in this schema show the chosen view over the system domain.

In Lange's methodology, the navigational design is done in the Composition Framework, which is part of the design phase of his multimedia design methodology [5]. The Composition Framework consists of Composition Identification and Composition Refinement. Composition is defined as a relationship, its classes and cardinalities, and the relationship's semantics. After creating a list of relationships, by applying the eliminating criteria, the unnecessary relationships (irrelevant and derived) are eliminated. The next step is to refine the compositions by defining the domain, cardinality, invariant and the link class semantics.

SOHDM [6] supports navigational design in two design phases, called the View Design and the Navigation Design phases. In the View Design phase, navigational units are created by reorganizing objects, and each navigational unit represents an Object-Oriented (OO) view. An Object-Oriented view is built on the basis of responsibilities and attributes in CRC cards, and the relationships in class structure diagrams. Object-Oriented views are divided into three types: base, association, and collaboration views. In the Navigation Design, access structure nodes (ASN) are created as new navigational units. ASN are determined from different scenarios. Although ASN and OO views are navigational units, they are different. ASN contains only access paths to OO views, but OO views contain the actual information that users want to obtain. The OO view and ASN are presented as nodes in navigational design. Moreover, the Navigational Link Matrix is used to represent all the links between nodes.

Adams' hypertext instructional design methodology includes navigational design within the "specifications stage" [8]. It converts the OOTD graph from the conceptual design phase into a Content Flowchart for implementation in a multimedia system. The first step in this stage is to find out the first and second order links in the OOTD graph, to build a System Flowchart. Then grouping the key concepts into sets of nodes and obtaining a new graph. Using Gagne's events of instruction the new graph is integrated to obtain a Content Flowchart. [12].

The Relationship Management Methodology (RMM) uses a technique called M-Slice to group information into meaningful information units [13]. Converting the M-Slice graph the Application Diagram is obtained to represent the navigational design. Comparing this Application Diagram with the pre-designed Application Diagram, a consistent final version of navigational design is obtained.

In HDM, the process of navigational design phase consists of defining the specification of a particular browsing semantic that is compatible with the given multimedia system. To achieve this, the designer must translate HDM application specifications into running applications, which means mapping HDM primitives into implementation structures of the target environment [14]. The minimal browsing semantic that is compatible with normal 'nodes to links' structures are called 'default browsing semantics'. This default browsing semantics assumes end users can view or read only one standard 'node' at a time, and this

node must be a unit. Different types of links are used to achieve the goal of specifying the navigational path for the multimedia system. There are concepts of 'abstract' and 'concrete' links. 'Abstract' links come from the abstract objects, either components or entities, and each 'abstract' link needs to be transformed into a set of 'concrete' links that connect between units.

RMD diagrams will be built in the navigation design by combining the E-R diagram from the conceptual design phase with the "entity" diagram where the entity diagram shows the structural and perspective relationships of the entities in the multimedia system [2]. Structural links are for connecting different instances of the same entity. Application links are used to represent domain dependent relationships among entities, or their components. Perspective links are used to connect units of the same component, which have different perspectives (i.e. Interview-text; Interview-video).

The Dexter model has a more detailed description of the links, splitting their contents into two parts. The first part specifies the source component and the second part specifies the target component [10].

AHM applies time schedule to the composite components to specify navigation patterns. This methodology uses synchronization arcs to define the duration between the related atomic/composite components [11]. Synchronization arcs are constraints that the implementation system should apply when presenting two or more components either concurrently or in series. AHM also allows the designer to specify the relative start time of each component (i.e. offset).

Abstract Interface Design Phase

The first language used in human-computer dialogue is input language, which is expressed via the physical actions performed by users, such as typing on a keyboard or clicking on a mouse [15]. The second one is output language, expressed in several ways like graphics on screen, audio through speakers, the systematic responses, or tactile sensations. Both languages are combined by two major components: "Meaning"(content) and "Form"(How the meaning is conveyed).

OOHDM provides the Interface Design phase, in which the interface objects perceived by the users and the way in which different navigational objects appear are defined [4]. The Abstract Data View (ADV) design approach can be used for describing the user interface of a multimedia system [16]. ADV is suitable for both, user interfaces and general interfaces. It has the structures and operators to separate the interface from the application and enhance the programmer's ability to maintain the separation during implementation for any programming environment.

RMD and RMM only support user interface design. RMD diagrams are needed for helping the user interface design in both RMD and RMM. While the final version of the navigational design has not been finished in RMM, the design of the user interface has begun. The final version of navigational design is the result of M-Slice, and the pre-designed Application Diagram. This is called Top-Down and Bottom-Up design approach [13].

Lange's multimedia design methodology has a design phase called "GUI Framework". There is an activity related to this framework called Presentation and Window Identification. The first task in this activity is to determine the application and browsers of the domain. The second task is to determine which presentation should be presented in every window [5].

SOHDM provides Page Schemas and some user interface (UI) components for designing the user interface [6]. These user interface components provide the appropriate notation for UI specifications.

AHM uses channels to support Abstract Interface design. Channels are abstract output devices for expressing the content of a component to the user. Associated with each channel are the default presentation characteristics for the media types presented via that channel, like speed of a video channel, or font size and style for a text channel. Each channel can store two different media-type items. The first one is media-type independent specification, such as background, foreground and highlight color, or whether the channel is on or off. The second one is a media-type dependent characteristic, like font size and style for text, or scaling factor for graphics. All these channels permit the interaction with users when mapped to physical devices [11].

Multimedia Methodologies Summary

The structure of multimedia design methodologies have been analysed in the above sections and it is evident that design can be divided into three phases which are conceptual design, navigational design, and abstract interface design. Each design phase was compare across different methodologies, and the commonalities and peculiarities between them were identified. This review has provided the impetus for the design and implementation of a multimedia system design toolkit that can assist designers, to plan, analyze, draft, and design multimedia systems. From our analysis, we found that the entity-relational and object-oriented design methods are the most common methods used in multimedia system design. Among the design methodologies presented in this paper, most of them have support for the conceptual and navigational design phases, however, half of them do not have sufficient support for the Abstract Interface Design phase. SOHDM and AHM seem to provide better support for the Conceptual Design Phase. OOHDM, RMM, SOHDM, HDM, Dexter, and AHM provide good support for the Navigational Design Phase. The Abstract Interface Design phase is supported by OOHDM and SOHDM. In the following section the design of our multimedia design toolkit is explained and its functionality introduced.

3. "MSDT" DESIGN MODEL

In this section, a software tool called a Multimedia System Design Tool (MSDT) is presented. This software tool, which was implemented in JAVA, can help with the design of multimedia system and was designed by analyzing the characteristics of several different multimedia design methodologies discussed in the previous section. MSDT uses entity-relational design methods to help with the design of multimedia system. We have chosen the entity-relational design method because most designers should have some knowledge of entity-relational modelling. From the analysis in previous section, it is observed that most of the multimedia design methodologies provide some support for Conceptual, Navigational, and Abstract Interface Design phases. Therefore, currently only these three design phases are supported by MSDT. However, the support provided for the Abstract Interface Design phase is still fairly rudimentary and requires further development.

MSDT Conceptual Design Phase

MSDT provides functions to create entities and relations. Users can create a conceptual model by using these functions. Since we have merged the techniques of the HDM entity definition and the AHM data component specification, then when users are creating entities, they can define the type of entity, as one of the

following: atomic, parallel composite, or choice composite. Parallel composite components present all of their component parts at the same time, but choice composite present at most one of their composition parts. Moreover, the presentation specification can give a more specific description of the media presenting the entity [11]. If an entity is defined as composite type (either parallel or choice), then MSDT will ask the user to enter the name of its composition entities. Furthermore, appropriate synchronization links will be drawn in the graph automatically if it is a parallel composite, indicating that those composition entities will be presented concurrently in the application's implementation. MSDT provides another functionality called Relationship Table. It is a table, which shows all the details of existing relationships. This feature provides the designer with a clear view of the multimedia system structure. The designer is able to delete relationships from this table by using the link number or the link name.

MSDT Navigational Design Phase

When the Conceptual Design Phase is finished and the Navigational Design Phase is started, MSDT will suggest that the users transform the relations into application links in the navigational model. This idea is borrowed from the automatic derivation of the application link function of HDM-tool [14]. Also, users can create structural links and perspective links for the navigation design, according to different needs in different situations. Application links represent domain dependent relationships among entities. Structural links are used to explore the information in different parts (i.e nodes) of the same entity. Perspective links are used to connect different perspectives (i.e. presentation) of the same component [9]. MSDT provides two functions for helping the users, namely the Navigation View and Link Table. The Navigation View can help designers to trace the navigation path, to ensure there is no dead end in the browsing of multimedia systems. Multimedia system may "get lost" during navigation and the Navigation View can help designers avoid this by facilitating link tracing. In the Link Table, all the links will be shown with their IDs, names, types, source and target entities [9]. The Link Table also provides the delete function, which can delete links by name or ID when required. These features of MSDT provide a good support for the Navigational Design phase. Since the Navigation View function lets the designers navigate the system virtually, designers can recognize and address the problems of “getting lost”, “dead end” and “loops” at earlier stages of design.

MSDT Abstract Interface Design Phase

The Abstract Interface Design phase is the last design phase supported by MSDT. In this phase, users can define the abstract interface between entities and users or corresponding applications. When designing abstract interfaces, two types of interfaces exist, namely, the abstract user interface and abstract general interface. The idea of providing this abstract interface design component is similar to ADV in OOHDM [4].

4. SOFTWARE PROTOTYPE EVALUATION

In this section the practical usage of the Multimedia System Development Toolkit (MSDT) is demonstrated. In order to do this, the software was applied to the development of a sample test multimedia application. This validation demonstrated the benefit of designing multimedia systems by using MSDT. The sample multimedia system developed is a hypermedia photo gallery. The application is intended to present photos of ruins across different countries to the general public, for access through the World Wide Web or on CD-Roms.

Development of Sample Photo Gallery

In the conceptual design phase, a model of the application domain was built using well-known entity-relationship modelling principles. First, the basic entities of the application and the relationships between them were characterized. This process specified the attributes and perspectives of the entities. The basic entities of the photo gallery are: Photo, Photo Description, Photographer and Interviews. The Photo entity contains a media type (e.g. image) and filename attributes. The Photo Description will display the description of the photos in text, and contains three attributes: title, description and place. The relationship between these two entities should be “photo description describes photo. Moreover, “photos are taken by photographers”. The Photographer entity includes attributes such as the photographer’s name, date of birth, nationality and personal description, and uses text to present these details. There are two types of photographer interviews, one is presented in video format and the other is presented in text. The Interview-Video entity presents the interview of the photographer, through video media and has the filename as attribute. The only difference between Interview-Video and Interview-Text, is the perspective. Interview-Text presents the interview in text and has the same attribute. The relation between photographer and Interviews is “grants”.

In order to develop a photo gallery application that can be accessed by the public, it is important to reorganize the information presented in the conceptual model. To achieve this, we need to associate the pieces of information by using different types of links. We can switch to navigation design mode by selecting “Navigational Design phase”, in the drop down list on the internal frame. A table, containing all the details of participating relations will then appear, which automatically suggests that the application links should be drawn. This will save the designers, time, while they consider the semantic relationships in the domain. Since we need to present a photo and its description together, we created a parallel-composite entity – Artwork. We choose the parallel-composite entity option, because the photo image and description text entities are presented simultaneously, and the choice-composite entity, only displays at most, one of the composite entities. The synchronization arc indicates the composition entities are presented concurrently and not in series, and the relationship “composite-by” shows the composite entity is combined by these entities. Next, we need to define the browsing semantics of the application, by connecting reasonable links between entities. There is one important point that needs to be kept in mind during linking, which is ensuring there is no dead end on the navigational path unless it is the final destination. During this design phase, we can traverse through the links by using our “Navigational View” tool to check the structure of the whole application.

The navigational structure of the Sample Photo Gallery is composed by six entities which are: Main Page, Country Content, Photographer, Artwork, Interview-Video and Interview-Text. The “Main Page” Entity presents a title, a list of countries, national flag images, and a list of photographers. This entity has two application links, connecting to Photographer and Country Content. The Country Content entity presents a list of photo previews and their titles, and it also contains two other attributes, which are country and filename. Country Content entity has one return link to Main Page and one link to Artwork. Since the

"Photo" entity is one of the composition entities of Artwork, the application link between "Photo" and "Photographer" should link to Artwork. Artwork has a structural link connected to itself, which means it can traverse to the next Artwork in the same country. Artwork also has an application link to Photographer, which can view the details of the Artwork owner. Moreover, Photographer is connected to Interview-Video and Interview-Text by two application links. Country Content and both Interview entities are also the same, with a return link to the Main Page. Both Interview entities can be treated identically, since they have a return link to Photographer and a perspective link connected between each other. The perspective link is used because both entities contain the same information but are presented in different perspectives.

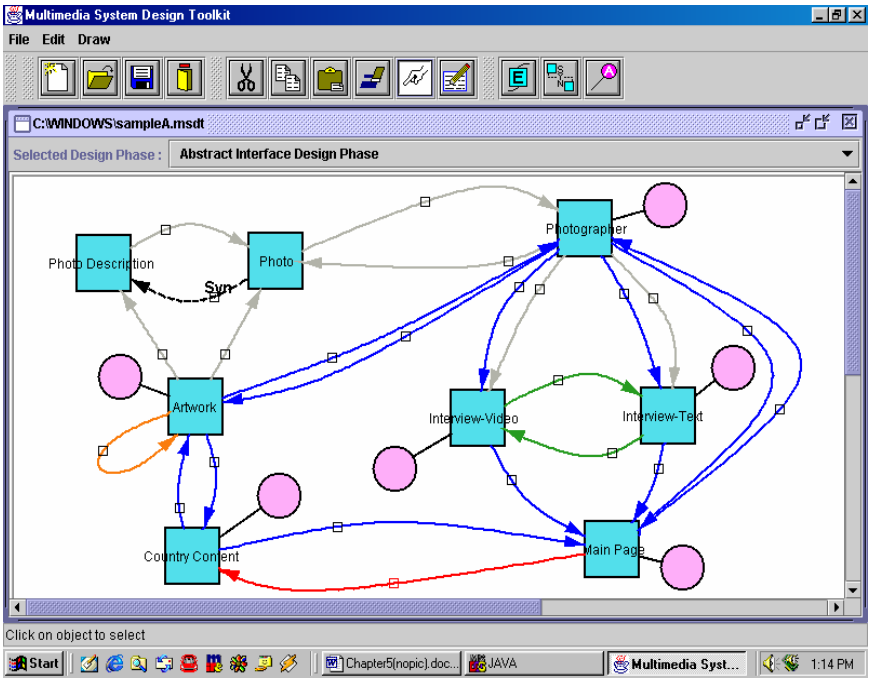

Figure 1. Completed MSDT model

After the navigational model for the Sample Photo Gallery has been defined, it must be made perceptible to the user through the application interface. In order to do this, an abstract interface model needs to be defined at this phase. This step defines which interface objects the user will perceive, the way in which different navigational entities will appear individually, and which interface objects will activate navigation. The "Main Page", is attached with an abstract interface, which specifies two interactions between user and sample photo gallery. When the user clicks on the country's name, then it will navigate to Country Content entity, which shows the preview of each Artwork. The other interaction is when the user clicks on the Photographer's name, which will activate the navigation to the photographer page, showing the details of the photographer and his work. There are two interactions in the abstract interface for Country Content. The first interaction triggers the navigation to the Artwork when the mouse is clicked on the title of the Artwork preview. The second one triggers the return to Main Page by clicking on the link named "Home". Artwork has three interactions in its abstract interface object, the first two are return to its Country Content and the link to the person who took this photo (i.e. "Photographer"). The last one is the linkage between the artworks taken in the same country, where the user can click on "Previous" to view previous artwork or click on "Next" to view the remaining artwork. Photographer has four interactions in its abstract interface, which are clicking "Home" hyperlink to return to Main Page, navigate to the Artworks of Photographer by clicking on the artwork titles, and two navigational paths to different interview representations by clicking on the "Interview-Video" or "Interview-Text" hyperlinks respectively. For the Interview-Text entity, the interaction box of the abstract interface indicates that clicking on "Home'' will activate the navigation back to Main Page. Clicking on the "Back" hyperlink can lead the user to return to the Photographer. The interview of photographer will be presented in video if the user clicks on "In Video" hyperlink. Basically, the Interview-Video entity has the same interactions in its abstract interface, but it has three more interactions for presenting the interview in video format. If the user clicks on the "Play" button, then the video will start playing from the start or resume from the pause position. Clicking on the "Stop" button will stop the video, and reset the time offset to 0. The video can be paused by clicking on the "Pause" button. The completed MSDT model is shown in Figure 1.

5. CONCLUSIONS

Multimedia system design is a complex activity. MSDT applies formal design techniques such as HDM, Dexter hypermedia model, AHM to improve the consistency of the system design and provide precise design guidelines. The development of the sample photo gallery shown above, demonstrates the success of MSDT in assisting the design of a multimedia system. The completed model of the application in MSDT has sufficient information for supporting the implementation. For example, a maintenance table of relations and links is provided in Conceptual and Navigational Design phases for the user to have a clear view of the structure and the relationship between different entities. Moreover, the navigational view feature can help the user to understand the browsing semantics of the application and fix the problems during the trial navigation. Other tools such as

Relation-To-Link is also a good tool in helping the designer to undertake the navigational design more efficiently. Clearly, MSDT cannot currently support the whole multimedia system design lifecycle. However, it can assist users to produce more consistent multimedia systems, and can provide a useful basic framework for creating a more complete multimedia design tool in the future.

A basic design was used to validate the usefulness of MSDT in the design of multimedia systems, namely the Photo Gallery Web page. This photo gallery shows the pictures of ruins, which were taken around the world by different photographers. It also presents an introduction of the photographers and their interviews. The conceptual design activities include defining the domain of the photo gallery, and specifying the set of entity and relationship definitions. In the navigational design phase, the browsing semantic and navigation of this application were tested by using the navigational view. The abstract interface design phase guides the designer to create the interactions between entities of the sample photo gallery. The completed sample photo gallery has a well-defined structure, which allows the user to perceive the application information data and media used. This application shows the benefits of using the MSDT to develop a multimedia system, and suggests that MSDT is a good option in assisting the design phases of the multimedia system development lifecycle.

Future research will improve the support offered by MSDT, with the aim of fully supporting the Abstract Interface Design and Implementation phases. A future extension will be to create an object-oriented version of MSDT.

6. REFERENCES

[1] M. Liu, C. Jones, and S. Hemstreet, "Interactive Multimedia Design and Production Processes", *Journal of Research on Computing in Education*, Vol. 30, No. 3, pp. 254-275.

[2] P. Balasubramanian, T. Isakowitz, and E. A. Stohr, "Designing Hypermedia Applications", *Proceedings of the 27th Annual Hawaii International Conference on System Sciences*, 1994.

[3] D. H. Jonassen, and H. Mandl, *Designing Hypermedia for Learning*, Germany, Springer-Veriag Berlin Heidelberg, 1990.

[4] D. Schwabe, G. Rossi, and S. D. J. Barbosa, "Systematic Hypermedia Application Design with OOHDM", *Proceedings of the ACM International Conference on Hypertext* (Hypertext'96), Washington: ACM Press, 1996.

[5] D. B. Lange, "An Object-Oriented Design Method for Hypermedia Information System", *Proceedings of the 27th Annual Hawaii International Conference on System Sciences*, 1994.

[6] H. Lee, C. Lee, and C. Yoo, "A Scenario-Based Object-Oriented Methodology for Developing Hypermedia Information Systems", *Proceedings of 31st Annual Hawaii International Conference on System Sciences*, 1998.

[7] B. Newman, "A methodology for design of large hypermedia systems", *IEEE*, 1998.

[8] S. A. Mengel, and W. J. Adams, "The Need for a Hypertext Instructional Design Methodology", *IEEE Transactions on Education*. Vol. 39, No. 3, August 1996.

[9] F. Garzotto, and P. Paolini, "HDM - A Model for the Design of Hypertext Applications", *Hypertext'91 Proceedings*, 1991.

[10] F. Halasz, and M. Schwartz, "The Dexter Hypertext Reference Model", *Communications of the ACM*. Vol. 17 No. 2, February 1994, p.p. 30-39

[11] L. Hardman, D. C. A. Bulterman, and G. V. Rossum, "The Amsterdam Hypermedia Model", *Communication of ACM*, Vol. 37, No. 2, Feb 1994, pp. 50-62.

[12] E. K. Cook, and E. J. Kazlauskas, "The cognitive and behavioral basis of an instructional design: Using CBT to teach technical information and learning strategies", *J. Educational Technology Systems*, Vol. 21, No. 4, 1993, pp. 287-302.

[13] T. Isakowitz, A. Kamis, and M. Koufaris, M., "Reconciling Top-Down and Bottom-up Design Approaches in RMM", *The DATA BASE for Advances in Information Systems*, Vol. 29, No. 4, 1998.

[14] F. Garzotto, P. Paolini, and D. Schwabe, "HDM - A Model-Based Approach to Hypertext Application Design", *ACM Transactions on Information Systems*, Vol. 11, No. 1, January 1993, pp. 1-26.

[15] A. Marcus, and A. V. Dam, "User-Interface Developments for the Nineties", *IEEE* 1991.

[16] D. D. Cowan, and C. J. P.Lucena, "Abstract Data Views, An Interface Specification Concept to Enhance Design for Reuse", *IEEE Transactions on Software Engineering*, Vol. 21, No. 3, Marc 1995.