Using UML To Model Presentational of Web Application

Mohd. Nazir Ahmad @ Sharif, R. Azmi Ahmad,
Syed Mohd Haidzir Syed Abd Rahman, and Nor Hidayati Zakaria

Faculty of Computer Science & Information System
Universiti Teknologi Malaysia, Skudai
81310, Negeri Johor Darul Takzim.
Malaysia.
{nazir, hidayati}@fsksm.utm.my
{edhai, azzmi}@hotmail.com

ABSTRACT

Web-based applications are currently become more complex system. The range of Web-based applications varies enormously, from simple static Web sites that are essentially hypertext document presentation applications, to sophisticated high-volume e-commerce applications often involving supply, sourcing, ordering, payment tracking and dispatch of goods or the provision of services. In other words, it provides users with business logic facilities, navigational structure, and also presentational aspects. In the context of presentational design, Web applications often contain significant multimedia content such as images, movie clips, sound clips and text, requiring specialist mechanism for their development. These elements of presentational need for mechanisms that can be used to support the modeling of user interfaces, which showing how navigational is presented to the user. We choose UML as modeling tool for presentational design, based on two important advantages. First, UML has been become a de facto standard for the design of object oriented systems. Second, and the most important one, CASE tools supporting UML can be used in the Web application development process. Note, UML can be used to model different kinds of systems such as software systems, hardware systems, and real-world organizations. The purpose of this paper is to show how UML notation and UML modeling techniques can be used in the Web application presentational modeling. UML stereotypes are specified for presentational design, according to the UML extension mechanisms defined in UML.

Keywords: UML, Presentational Design, Web Application, UML Extension.

1.0 Introduction

The complex Web applications provide users with functionality to manipulate information. Currently, the most Web applications development focuses on the business logic implementation and paying little attention to the navigational structure and presentational design. Existing in fact, quality of a Web application depends not only on the implementation of business logic of the system, but also need well-structured navigational and presentational.

The aim of this paper is to evaluates and shows how UML modeling can support the analysis and design phases, particular focuses on presentational design of Web applications. UML stereotypes are specified in [1], [6], [7] with some modifications will be used for modeling user interface elements of Web applications. These specified stereotypes according to the mechanisms for standard extensions defined in the UML. This model-based approach by Hennicker and Koch in [1] is a methodology that allows for a partially automated generation of templates for Web applications. This approach makes a clear distinction between the conceptual design of the domain, the design of Web navigational structure and the presentational design.

The presentational design supports the modeling of an abstract user interface showing how the navigational of application is presented to user. Presentational design means defining the way how application will be explored, selecting user interface objects to activate navigation and determining which interface transformations will take place.

This paper is organized as follows: Section 2, provides an overview of the basic concepts of UML modeling techniques that can be used in the analysis and design of Web applications. Section 3, explains a concept of presentational design, which the scope of this paper will focus on static presentational design. Including in this section is a set of stereotypes proposed for the basic modeling concepts used in the presentational models. In section 4, a simple example of presentational design is given. In the last section, some concluding remarks and an overview of future work is outlined.

2.0 UML Basic Concepts

UML (Unified Modeling Language) is a standard object-oriented design language that has gained virtually global acceptance among both tool vendors as well as software developers. UML has been standardized by the Object Management Group (OMG). To understand UML, one must important is to have an understanding of Object Oriented Programming (OOP). OOP is based on classes and objects. A class can be viewed as a blueprint of the attributes and behaviour of a part of a program.

The most important concepts in UML are about actor, use case, use case diagram, sequence diagram and class diagram. First, the actor of the system is an entity outside the system that interacts with use cases [3]. Second, the use case which focuses on the requirements of a system rather then the way a system will actually be designed. We need to identify use cases of the system and then draw a use case diagram, which the diagram shows the relationships among actors and use case within a system. UML provides two most important relationships that can be used to structure use cases. These are *include* and *extends*. An include relationship between two use cases means that the sequence diagram of behaviour described in the included use case is included in the sequence of the base use case. Including a use case is thus analogous to the notion of calling a subroutine [5]. The extends relationship provides a way of capturing a variant to a use case. Typically, extensions are used to specify the changes in steps that occur in order to accommodate an assumption that is false [5]. The extends relationship includes the condition that must be satisfied if the extension is to take place, and references to the extension points which defines the locations in the base use case where the additions are to be made. The use case and use case diagram become a valuable tool when communication with non-programmers such as customers and marketing executives [4]. Thus, it provides a high-level understanding and means of describing program requirements without knowing how the implementation of the program. Third, the sequence diagram shows object interactions arranged in time sequence. In particular, it shows the objects participating in the interaction and the sequence of messages exchanged. It includes time sequences but does not include object relationships. The sequence diagram is constructed from vertical lines representing objects in time and horizontal arrows representing information being passed from one object to another. Lastly, is about a class diagram which represents how the blueprint of the system. It that shows a collection of static models elements, such as classes, types, and their contents and relationships.

UML provides a general extension mechanisms that can be used to defines specific stereotypes to model specific aspects of the system such as conceptual, navigational or presentational models. UML extension is a predefined set of Stereotypes, Tagged Values, Constraints and notation icons that collectively extend and tailor the UML for a specific domain or process [3]. The UML is broadly applicable without extension, so companies and projects should define extensions only when they find it necessary to introduce new notation and terminology. There have been a number of initiatives to extend UML to new application areas, including Web systems and databases. Some of these efforts have been carried out as OMG initiatives, while individual researchers and individual companies such as Rational have done others.

3.0 Presentational Design

The part of this methodology for the analysis and design of Web applications is to model presentational aspects of the Web applications. The development of Web applications requires abilities to choose multimedia contents, to define an adequate structure, to design the user interface and to select the appropriate implementation technique [6], [11]. The models proposed in [1,6] are built with well-known UML model elements and extensions [3]. These extensions are defined following the UML extension mechanisms.

The presentational design supports the modeling of an abstract user interface describing how the navigational will be presented to the user. To achieve this purpose, a static and dynamic presentational model is built. Anyway, the scope of this paper will demonstrate how this approach can be used to model static presentational model. The static presentational model can be represented by UML composition diagrams [1] that describe how the user interfaces are built. However, this paper will propose the static presentational modeling represented by UML class diagrams. The class diagrams can show explicitly the static presentational models and it is well-known and among the most important diagrams that supporting by many CASE tools. Modeling presentational includes modeling a user interface of Web applications. The user interface object can be either a primitive user interface object like text, image and button, or a composition of user interface objects. Thus, for the most frequently user interface objects the approach [1] define stereotypes according to the extension mechanism provided by UML. These examples of user interface objects are *text, image, audio, video, form, and button.*

This UML profile, which based on [1], [6], [7], defines a set of stereotypes that can be used in the development of static presentational models of Web applications. These stereotypes are *presentational class, frameset, frame, text, anchor, button, image, audio, video, form, collection* and *anchored collection*. However, in this paper we will improve the stereotypes by on refining the stereotypes *anchor* as *text link* and *icon link*. These new definition of anchor stereotypes are more specific, explicit and they are common situations terminology used in Web applications user interface. Besides that, we also refining semantic of *button* and introduce new stereotypes are called *table, text link collection,* and *icon link collection*. As the results, a set of new and previous stereotypes are defined as follows; *presentational class, frameset, frame, form, images, audio, video, collection, text, button, text link, icon link, table, text link collection,* and *icon link collection*. For each stereotype of user interface objects have the following semantics:

- A *presentational class* models the presentation of navigational class. Instances of a presentational class are containers, which comprise modeling elements like texts, images, video, audio, and collections.
- A *frameset* and *frame* is a top-level element, which is modeled by a composite that contains (lower level) presentational objects but may also contain an arbitary number of nested framesets. An area of the frameset is assigned to each lower level element – so called frame.
- A *form* is used to request information from the user who supplies information in one or more input fields or selects option from a browser or checkbox.
- *Image, audio* and *video* are multimedia objects. An image can be displayed; audio and video can be started, stopped, rewinded and forwarded.
- A *collection* is consists of a set of user interface objects. Meaning that, a collection can have more than one user interface objects like text elements, text link, and icon link. This model element can provide a convenient representation of a collection of user interface objects.
- A *text* is a sequence of characters together with formatting information.

- A *button* is a clickable area, which has an action associated to it. This paper refines that button here is a primitive button and not an icon button.
- A *text link* is a clickable text area, which is the starting point of a navigation establishing this way the relationship to other links.
- An *icon link* is a clickable icon area, which is also the starting point of navigation. Icon is a small picture used in place of another entity. For example, most Web applications use icons to represent another view or links of a windows or view of applications.
- A *table,* is a frame for collection of user interface objects. In Web applications, we always put user interface objects such as image, text, and button inside the table. We also can use table for better view, design and arrangement for user interface object locations.
- A *text link collection* consists of a set of *text links.* This model element introduced to provide a convenient representation of composites that are frequently used. In this context, it provides a convenient to represent a collection of text links.
- An *icon link collection* consists of a set of *icon links.* This model element also introduced to provide a convenient representation of composites that are frequently used. In this context, it provides a convenient to represent a collection of icon links.

4.0 Example Application

As an example to illustrate the UML approach presented in this paper, some of the static presentational models of Web applications called e-BulletinIS are presented. This online bulletin provides information about department, articles that can be read, download, simple administration features, and announcements facilities. Figure 1.0 as below, illustrates a part of Web page of the resulting this application. This Web page called *Home* is a presentational class as shown with icon-based defined in [1], [6], [7] and this paper.

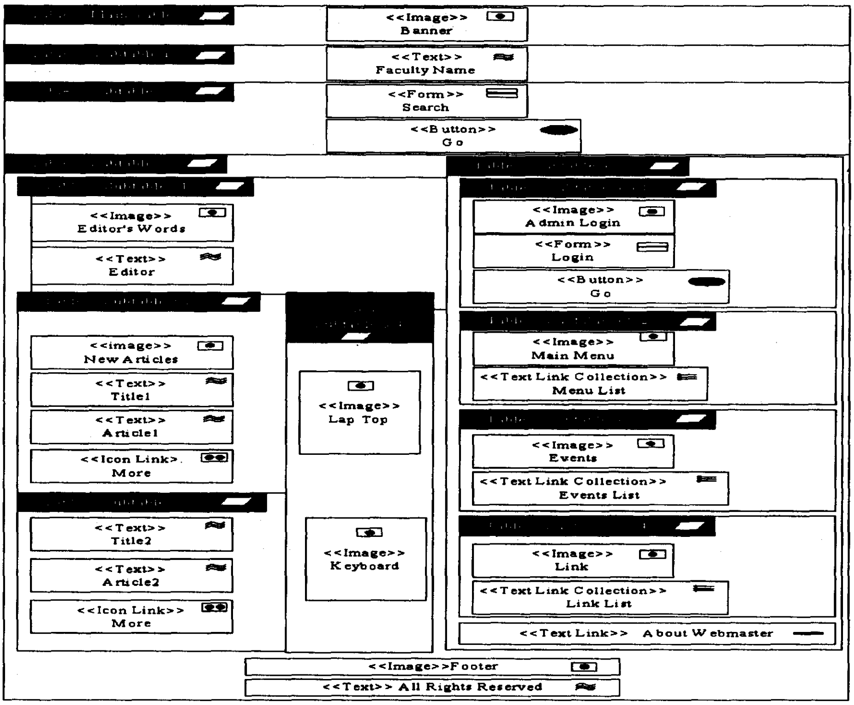
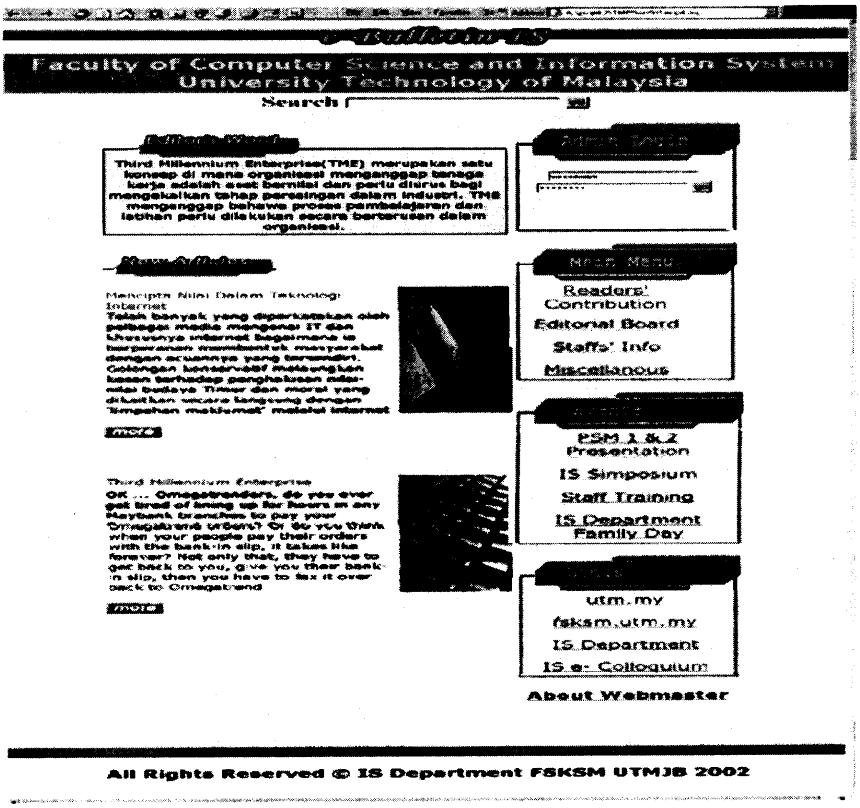


Figure 1.0: The Icon-based *Home* Web Page

The Web page is described through a main table and sub-tables where elements of user interface objects like image, text, icon link, text link, and form are put inside the tables (see Figure 1.0). In Figure 2.0, the real *Home* Web page is delivered by using stereotypes of user interfaces objects that have presented here. Figure 3.0 demonstrates how this Web page can be modeled and represented using simple UML class diagram. The Web page consists of user interface objects like image, text, and form. These user interface objects have their own attributes that so they can be modeled and presented as a class. These classes are composited from the Web page through composition relationships provided in UML, where it is illustrated by diamond line with always 1 cardinality (see Figure 3.0). The composition relationship is a strong aggregation, meaning that, component classes are existence-dependence on their composite class. Exactly, Figure 3.0 shows a part of the main UML composition for *Home* Web page.



e-Bulletin IS

Faculty of Computer Science and Information System
University Technology of Malaysia

Search

Third Millennium Enterprise(TME) merupakan satu konsep di mana organisasi menganggap tenaga kerja adalah aset bernilai dan perlu diurus bagi mengekalkan tahap persaingan dalam industri. TME menganggap bahawa proses pembelajaran dan latihan perlu dilakukan secara berterusan dalam organisasi.

Mencipta Nilai Dalam Teknologi Internet

Telah banyak yang diperkatakan oleh pelbagai media mengenai IT dan khususnya internet bagaimana ia berperanan membentuk masyarakat dengan acuannya yang tersendiri. Golongan konservatif melaungkan kesan terhadap penghakisan nilai-nilai budaya Timur dan moral yang dikaitkan secara langsung dengan "limpahan maklumat" melalui internet

more

Third Millennium Enterprise

OK ... Omegatrenders, do you ever get tired of lining up for hours in any Maybank branches to pay your Omegatrend orders? Or do you think when your people pay their orders with the bank-in slip, it takes like forever? Not only that, they have to get back to you, give you their bank-in slip, then you have to fax it over back to Omegatrend

more

Main Menu

Readers' Contribution

Editorial Board

Staffs' Info

Miscellanous

PSM 1 & 2 Presentation

IS Simposium

Staff Training

IS Department Family Day

utm.my

fsksm.utm.my

IS Department

IS e- Colloquium

About Webmaster

All Rights Reserved © IS Department FSKSM UTMJB 2002

Figure 2.0: The *Home* Web Page

UML class diagram as shown in Figure 3.0 to Figure 6.0 represents static presentational modeling with UML. Briefly, The *Home* is represented as presentational class, which constructed by one main table. This main table is composited to three table components, where each table will be composited to their own user interfaces objects (see figure 3.0 to 6.0). For an example, Figure 3.0 shows a main table is a composition of *Banner* image, *Subtable1, Subtable2* and *Subtable3*. These tables own their user interface objects like text, image, form, icon link, table, and text link; so, they can be shown as classes. Thus, we can put them with some attributes stereotypes such as size, font, and colour for examples for text.

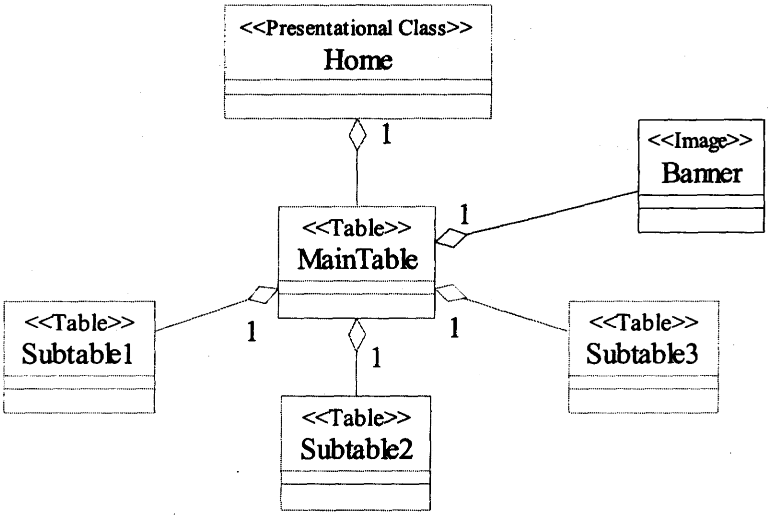


Figure 3.0: UML Composition For *Home* Web Page

Similarly, Figure 4.0 as below is a UML composition for tables named *Subtable1* and *Subtable2*. For each table owns their user interfaces objects such as text and form. Examples, *Subtable2* owns form class named *Search*. This form has a stereotype attribute called *button* and its name as *Go*. This form provide interface to perform login verification for administration admission. We can also model others elements of form stereotypes attributes such as *button, combo box, input text,* and *submit* by using stereotypes mechanism in UML.

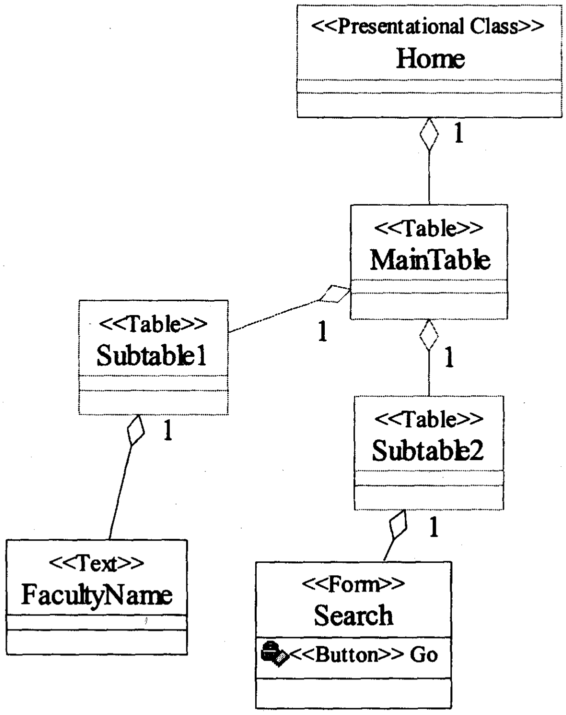


Figure 4.0: UML Composition For *Subtable 1* and *Subtable* 2

Figure 5.0 illustrates similarly the composition of user interfaces objects for *Subtable3*. This table consists of tables named *Subtable3.1, Subtable3.2, Subtable3.3, Subtable3.4,* and *Subtable3.5*. We can see the composition of table named *Subtable3.3* and *Subtable3.5.1* in Figure 6.0.

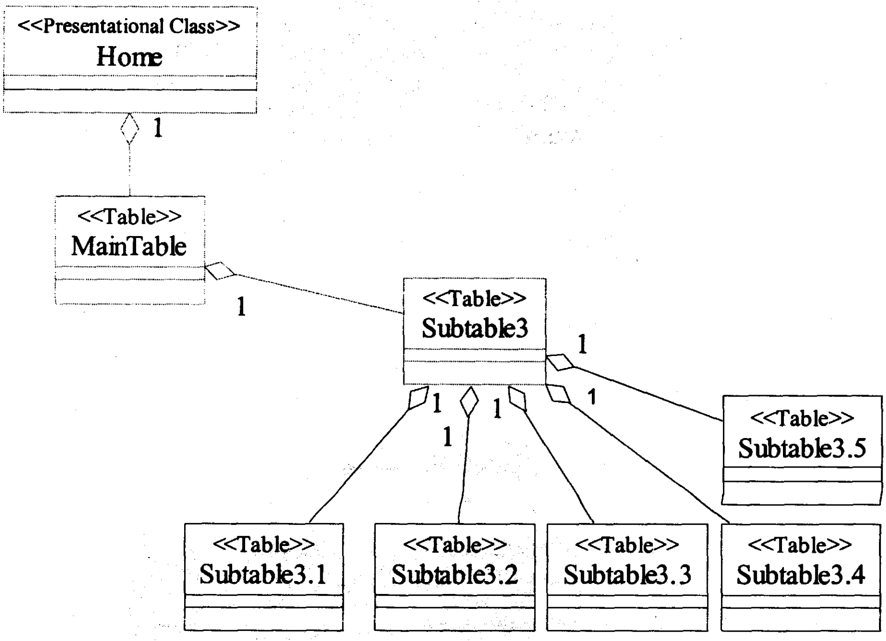


Figure 5.0: UML Composition For *Subtable 3*

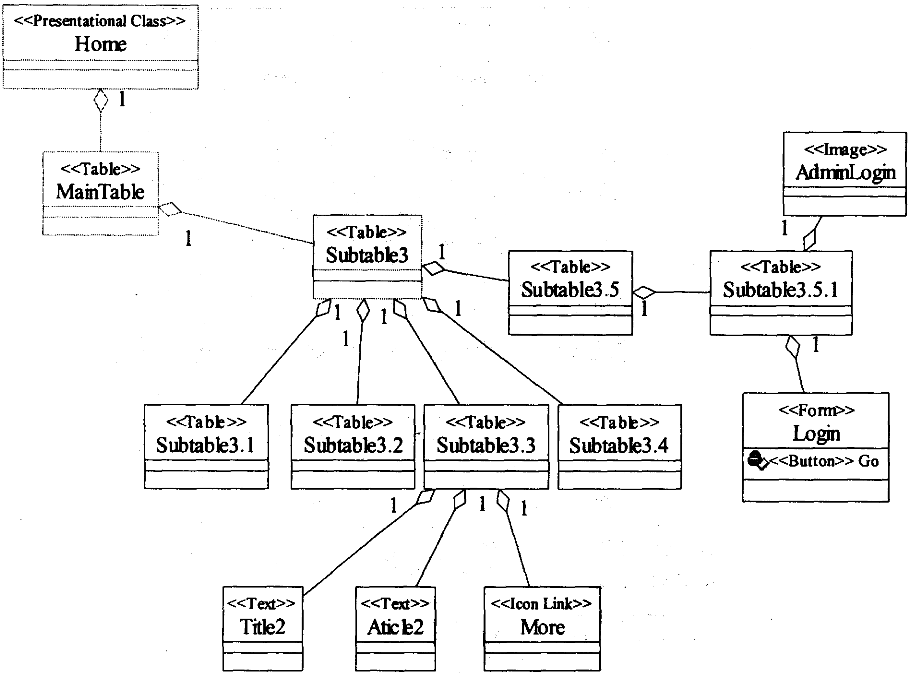


Figure 6.0: UML Composition For *Subtable 3.3, Subtable 3.5.1*

These UML class diagrams have been constructed using Rational Rose 2000, and the application has been developed by using developer tools, Php programming and Access 2000 database.

5.0 Conclusions

In this paper we have presented a step for modeling static presentational design of Web applications. This is part of the UML-based methodology described in [1], [6], [7] for modeling Web applications. The modeling is presented here is based on UML profile, in particular references to the UML standard extension mechanisms. The mechanism used is the definition of stereotypes. There have been a number of initiatives to extend UML to new application areas, including Web systems and databases. Using UML as modeling language has the advantage of using a well-known standard, and therefore the development may be supported by existing CASE tools. This paper also proposes on refining of previous stereotypes in [1], [6], [7] and testing the stereotypes introduced in [1], [6], [7] through modeling simple Web application prototype called e-BuletinIS.

Our future work will concentrate on refining the models here presented and also will concentrate on testing and evaluating navigational design introduced in [1], [6], [7]. We also to evaluate the approach in [8], [9], [10], [12] and try to come up with combination approach in order to model Web applications.

Acknowledgment

We would like to thank Dr. Nora Koch for her helpful opinions and advices through e-mail disscussion.

References

[1] Baumeister, H., Koch, N. and Mandel L. "Towards a UML Extension for Hypermedia Design". In Proceedings UML'99 Conference, France,R., Rumpe,B.(Eds), LNCS, Vol.1723, Springer Verlag, 614-629. 1999.

[2] Williams, Tom, "Standard Graphical Notation Proposed For Object-Oriented Language", Electronic Design, Vol.45, No.2, pp.137-138. 1997.

[3] G.Booch, J.Rumbaugh, and I.Jacobson. "Unified Modeling Language User Guide". Addision Wesley. 1999.

[4] Douglass, Bruce Powel, "Designing Real-Times Systems With the Unified Modeling Language", Electronic Design, Vol.45, No.20, pp.132-138. 1997.

[5] Coleman, Derek."A Use Case Template: Draft for discussion". Fusion Newsletter, April 1998. http://www.hpl.hp.com/fusion/md_newsletters.html.

[6] Baumeister, H., Koch, N., Hennicker, R. and Mandel L. "Extending UML to model Navigation and Presentation in Web Applications". Technical Report 9901, Ludwig-Maximilians-Universitat Munchen. March 1999.

[7] Hennicker, R. and Koch N. "A UML-based Methodology for Hypermedia Design". UML'2000. 2000.

[8] Conallen, J. "Modeling Web Application with UML". Conallen, Inc. Object Oriented Application Development. Ogos 2000. http://www.conallen.com/ModelingWebApplications.html

[9] Conallen, J. "Building Web Applications With UML". Addison Wesley. 2000

[10] Conallen, J. "Modeling Web Applications Architectures With UML". Communication Of The ACM Vol 42, No 10. Oktober 2000.

[11] Koch, N., & Luis, M. "Using UML to Design Hypermedia Applications" Institut fur Informatik, Ludwig-Maximilians-Universitat Munchen. March 1999.

[12] Conallen, J. "Modeling ASP Applications with UML". Principal Consultant, Conallen, Inc. 2000.