Modeling the User Interface of Web Applications with UML

Rolf Hennicker[1], Nora Koch[1,2]

[1]Institute of Computer Science
Ludwig-Maximilians-University Munich
Oettingenstr. 67
80538 München, Germany
{kochn,hennicke}@informatik.uni-muenchen.de

[2]F.A.S.T.
Applied Software Technology GmbH
Arabellastr. 17
81925 München, Germany
koch@fast.de

Abstract. Today's User Interfaces (UIs) are complex software components, which play an essential role in the usability of an application. The development of UIs requires therefore, not only guidelines and best practice reports, but also a development process including the elaboration of visual models and a standardized notation for this visualization.
Based on a UML extension for Web applications of previous work, we are focusing in this paper on the user interface design of Web applications, describing in detail the methodological steps – part of the UWE design process – to transform a model of the navigation into a user interface model. We propose a UML profile for the Web, which supports sketching and storyboarding (techniques that are widely used by UI designers without a precise notation). The strength of the UML models presented is given by the fact that they provide a precise notation and can be used as a basis for a semi-automatic generation of UI templates for Web applications.
Keywords. User Interface Design, Web Engineering, Development Method, UML Profile, Systematic Software Development, Web Applications

1 Introduction

Designing user interfaces for Web applications is a complex task which needs, in the same way as other aspects of system development, methodological guidelines and modeling techniques. In previous work (cf. e.g. [HK00]) we proposed a design methodology for Web applications which consists of conceptual, navigation and presentation design. These steps are part of the UML-based Web Engineering Approach (UWE). UWE is based on the Unified Modeling Language [UML01] and on the Unified

Process [JBR99]. The UML is extended to model Web applications and the Unified Process is tailored for the development of these applications (cf. [Ko00]).
The objective of the conceptual design is to build a conceptual model of the application domain taking into account the functional requirements captured with use cases. Traditional object-oriented techniques are used to construct the conceptual model, such as finding classes and associations and defining inheritance structures. The conceptual model is represented by an ordinary UML class diagram.
Based on the conceptual model the navigation design proposes a set of guidelines to construct a navigation model which represents the navigation space and the access elements that can be used for navigation (see Figure 1). We briefly summarize a set of stereotyped modeling elements for navigation design, like indexes, guided tours, queries and menus.
The presentation modeling aims at the design of abstract user interfaces and the design of the user interaction with the Web application. It consists of two steps: storyboarding user interfaces and building the presentation model which provides the source for an implementation. In contrast to [HK00] we focus in this work on the precise specification of the steps needed for a systematic design of the presentation. For this purpose stereotyped modeling elements which are suited for storyboarding and for modeling the concrete design of the presentation are included in our Web Profile. Our goal was also to show how these stereotypes can be used together with the WAE stereotypes of Conallen [Cj99] such that both approaches can benefit from each other.
The first step in the presentation design defines user interface views which sketch the content and the look and feel of the nodes. These user interface views can then be combined to storyboarding scenarios that are an excellent basis for developing a presentation model and constructing a pure HTML prototype. Sketching and storyboarding are common techniques of user interface designers, but usually they don't have a unique and precise notation [Pj94,Sb98]. Our approach provides methodological guidelines and modeling elements for storyboarding.

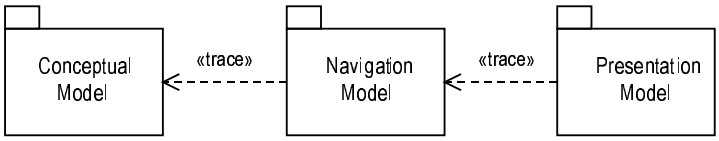

Fig. 1: Models of the UWE design process

The next step of our method goes towards an implementation (Figure 1). Based on the storyboard model the designer can decide whether he wants to choose a multiple-window technique and/or whether he wants to use frames and/or dynamic page generation. The objective of the presentation model is to show *where* the user interface views of the storyboard model are presented to the user, i.e. in which frame or window they are displayed. This also shows which contents are replaced when the user interacts with the system. In addition to the client pages, constructed in accordance with the user interface views, server pages in the sense of Conallen are identified when they are required for the dynamic generation of client pages.

Our approach has been validated with some case studies, for example the design of the Web-based conference review system. This case study is described in detail in [KKH01]. Related to our work are the UML extensions for interaction design provided by UML*i* and Wisdom. The UML*i* approach [PP00] defines different graphical representations for domain elements and interactive elements. It introduces notations for containers, inputters, displayers, editors and action invokers. These elements are used for the design of the abstract presentation without modeling the relationship between them. The Wisdom approach [JFC00] defines UML stereotypes to support task oriented development of user interfaces.
This paper is structured as follows: Section 2 summarizes the design steps of the UWE approach presented in previous work, i.e. the design of the conceptual model and the navigation model. Section 3 gives guidelines to build sketches (abstract user interfaces) with UML and how to use them to build storyboard scenarios. In Section 4, the presentation model is constructed showing where pages of a Web application are presented to the user and how the user interacts through this interface with the application. Finally, Section 5 presents some concluding remarks and an overview of future work.

2 Starting with the Navigation Model

Our method supports the design of Web applications building conceptual, navigation and presentation models. Conceptual modeling of Web applications does not differ from conceptual modeling of other applications.
Navigation design is a critical step in the design of Web applications. Even simple applications with a non-deep hierarchical structure will become complex very soon by the addition of new links. Additional links improve navigability on the one hand but imply, on the other hand, higher risk to loose orientation. Building a navigation model is helpful not only for the documentation of the application structure, it also allows for a more structured increase of navigability.

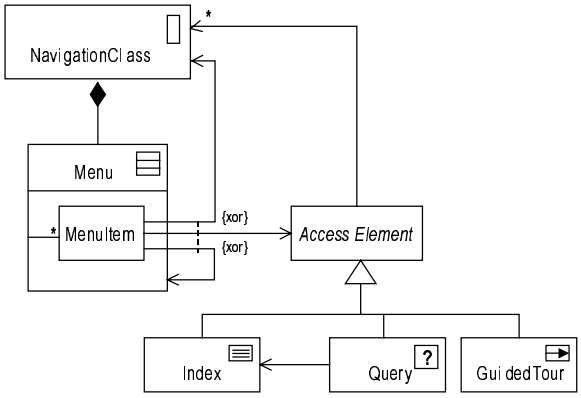

Fig. 2: Metamodel of the Navigation Modeling Elements

2.1 Modeling Elements

For the construction of the navigation model a set of stereotypes are defined to support a more intuitive modeling of the Web application. The stereotyped classes are the ´navigation class[a] and a set of access elements, such as ´index[a], ´guided tour[a], ´query[a] and ´menu[a], which are introduced and semantically explained in [BKM99,HK00]. The directed associations expresses ´direct navigability[a]. Figure 2 defines the well-formedness rules of the navigation modeling elements in terms of a metamodel. The class names occurring in this metamodel (with exception of the abstract class access element) will be used as stereotypes when concrete navigation models are constructed.
For instance, a particular index class can be defined as depicted in Figure 3 (left). For this we will often use the icon notation shown in Figure 3 (right).

Fig. 3: Access Element Index

In particular, instead of the notation for menus in Figure 4 (left) we will use as a shorthand notation the icon shown in Figure 4 (right) (and similarly for menus with more than two menu items).

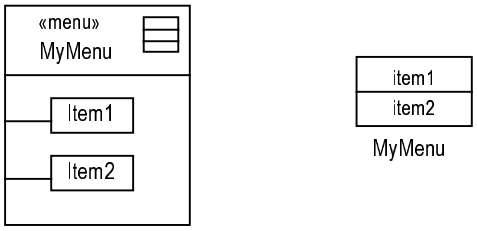

Fig. 4: Access Element Menu

2.2 Example

As an example to illustrate the design process we use the Web Site of a service company. This Web site offers information about the company itself, the departments, employees and projects and their relationships. We restrict ourselves in the example to these concepts, although many other aspects could be included (such as information about products, documents, events, press releases and job offers). Figure 5 shows a navigation structure model for this Web Site.

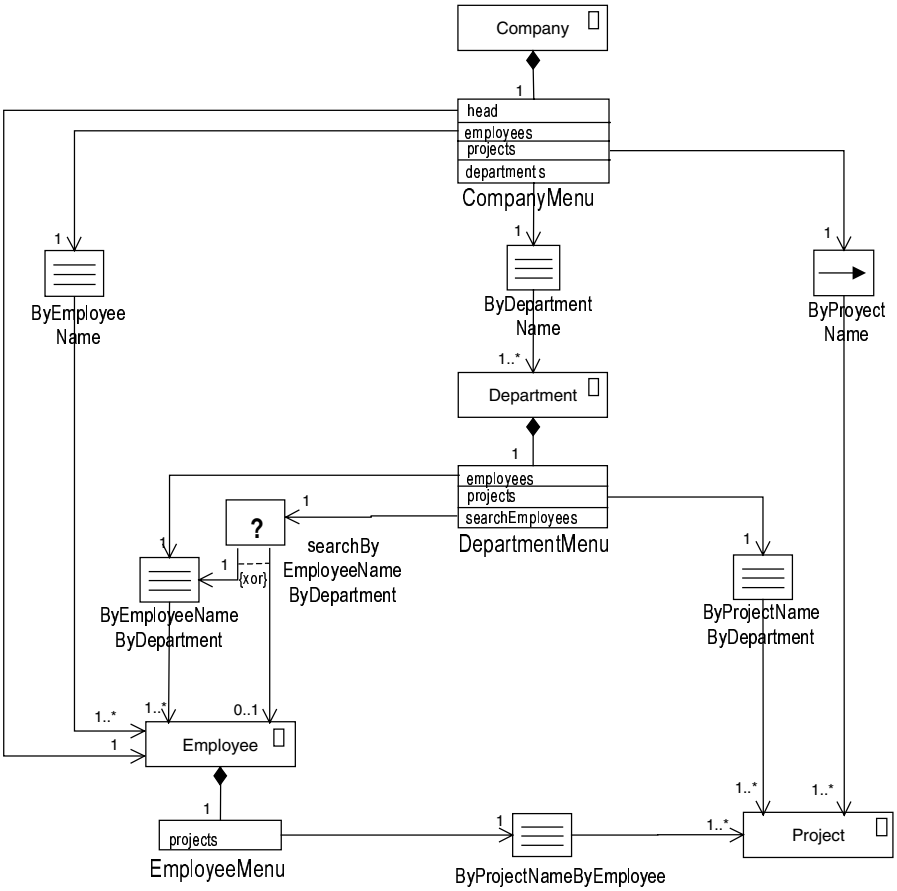

Fig. 5: Navigation Model

2.3 The Method

Our methodology for the design – part of the UWE approach – presented in [HK00] identifies a set of steps and guidelines for the construction of the navigation model. The main steps are summarized as follows:

(1) Classes of the conceptual model that are relevant for the navigation are included as navigation classes in the navigation model (i.e. those classes that are not a visiting target in the use case model are excluded).

(2) Required information on the omitted classes can still be kept as attributes of other classes in the navigation model. All other attributes of navigation classes map directly to attributes of the corresponding conceptual classes.

(3) Often additional directed associations are added for modeling direct navigability (thus avoiding navigation paths of length greater than one). Scenarios described by the use case model give the input for the choice of direct navigation.
(4) Access elements (indexes, guided tours, queries) are chosen to realize the navigation. In particular, role names at the directed ends of associations are becoming menu items of menus attached to navigation classes.

3 Storyboarding User Interfaces

The objective of the user interface modeling is to provide a notation and a guidance for the construction of a user interface visual model that shows the content and the structure of the single nodes (i.e. how each node is presented to a user). First, the Web designer proposes a sketch of each main user interface view. These are rough drawings of a couple of relevant elements of each navigation node. This sketching technique is frequently used by Web designers, but without having a precise notation for it as described by Sano [Sd96].
We propose to use an appropriate extension of UML for this purpose. It is mainly an *abstract* user interface design as it models only the structural organization of the presentation, given by interface objects, such as texts, images, forms and menus, and *not* the layout characteristics, in terms of fonts, colors, special formats, etc. Such decisions are taken typically during the development of a user interface prototype or in the implementation phase. The abstract user interface model may, however, provide some hints, for example, on the position and the size of the interface objects relative to each other.
In order to construct an abstract user interface model, one has to decide, on the one hand, which user interface objects will be used for the presentation of the instances of navigation classes and, on the other hand, which ones will be used for the presentation of the access elements.
The abstract user interface design may be considered as an optional step as the design decisions related to the user interface can also be taken during the realization of the user interface. However, the production of sketches of this kind is often helpful in early discussions with the customer.
After having designed the different user interface views storyboarding scenarios can be developed which show sequences of user interface views in the order in which a user can navigate from one view to another [Pj94]. This aids in visualizing the organization of the Web application structure in a more intuitive manner than the navigation model does. Both, the sketches of user interface views as well as the storyboarding scenarios are a useful means for the communication between a customer and a Web designer. In particular, they can be validated w.r.t. the use cases identified during an analysis phase.

3.1 Modeling elements

For the construction of the abstract user interface we propose the modeling elements shown in Figure 6. As for the navigation elements in Section 2, each class in Figure 6 defines a stereotype which will be used in concrete user interface models. The associations and the inheritance relation show again the well-formedness rules; the notation of the interface elements in form of icons stem from Baumeister, Koch and Mandel [BKM99].

User Interface View. A user interface view specifies that each instance of this class is a container of all the abstract user interface elements which are presented simultaneously (i.e. at one moment in time in one window) to the user. For user interface view classes we use the stereotype ´UI view[a] and its corresponding icon depicted in Figure 6.

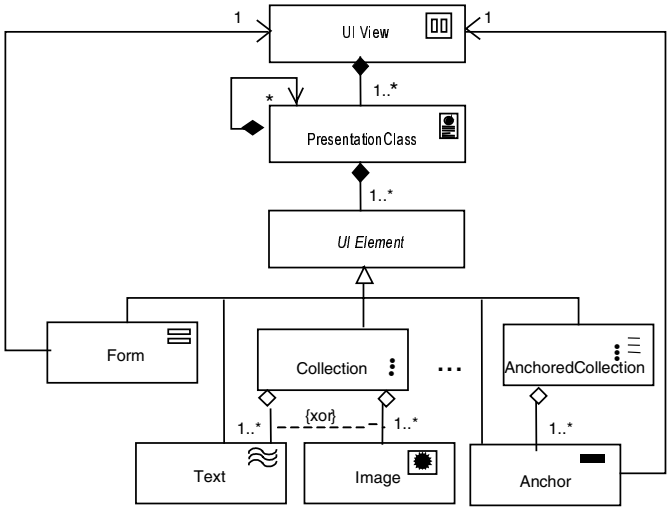

Fig. 6: Metamodel of the Abstract User Interface Elements

Presentation Class. A presentation class is a structural unit which allows to partition a user interface view into groups of user interface elements. For presentation classes we use the stereotype ´presentation class[a] and its corresponding icon depicted in Figure 6.

User Interface Element. A user interface element is an abstract class which has several specializations describing particular interface elements (Figure 6)[1].

For instance, the stereotyped classes ´text[a], ´image[a], ´video[a], ´audio[a], ´anchor[a], and ´form[a], are subclasses of user interface element for modeling texts, images etc.. The classes ´collection[a], and ´anchored collection[a] are also subclasses of user interface element which provide a convenient representation of frequently used composites. Anchor and form are the basic interactive elements. An anchor is always associated with a link for navigation. Through a form a user interacts with the Web application supplying information and triggering a submission event.

[1] We neither discuss here particular details of the user interface elements nor their mapping to concrete GUI widgets.

3.2 Example

Figure 7 shows a UI view that is part of the abstract user interface model of the service company. It is the composite of the presentation of the company's content (showing a picture of the company etc.) and the company's selection menu (which in turn is composed by four anchors.)

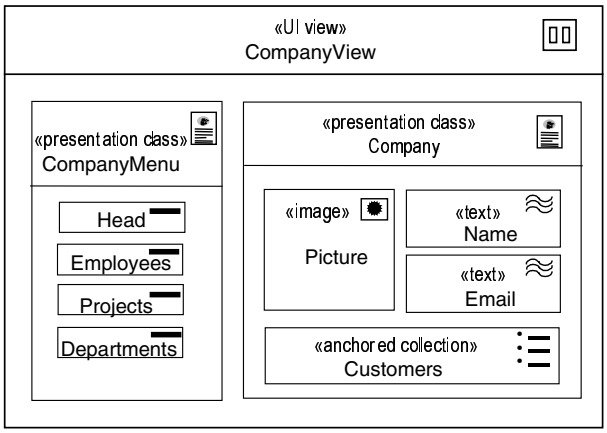

Fig. 7: User Interface View for a Company

Figure 8 shows a storyboarding scenario which describes the navigation to find the information about the head of the company and to find one department of the company out of a list of departments.

3.3 The Method

To model an abstract user interface we assume given a navigation model of the application. Each abstract user interface model is represented as a composition of classes. The following rules can be used as guidelines for the construction of the abstract user interface model:

(1) Construct a presentation class for each navigation class occurring in the navigation structure model. The presentation class defines a template suited to present the instances of the class by taking into account the given attributes. Stereotyped interface elements, such as «text», «image», «audio», «video» are used for attributes of primitives types and «collections» are used for lists, etc. Figure 7 shows the presentation class for a company.

(2) Construct a presentation class for each menu and index occurring in the navigation structure model. The presentation of a menu or an index class consists usually of a list of anchors. Use stereotypes «anchor» or «anchored collection» for this purpose. An example for the presentation of a menu is the CompanyMenu in Figure 7.

(3) Construct a presentation class for each query and guided tour. For queries use a «form» stereotype and for guided tours use a menu with items "next" and "previous" (which make it possible to navigate to the next and to the previous object within a guided tour).

(4) Determine which presentation elements should be presented together to the user (in one window). The corresponding presentation classes must be composed in a user interface view (stereotyped by ´UI view[a]). Since the user needs always a combination of conceptual data and navigation facilities, typically a user interface view consists of the presentation class constructed for a navigation class and of a presentation class constructed for a corresponding menu class (see CompanyView in Figure 7).

(5) Construct storyboarding scenarios represented by sequences of user interface views. For this purpose introduce links that connect an anchor (within a UI view) with another UI view thus showing the possible flows of presentations that can be caused by user interactions. An example for a storyboard model is the scenario for Head and Department shown in Figure 8.

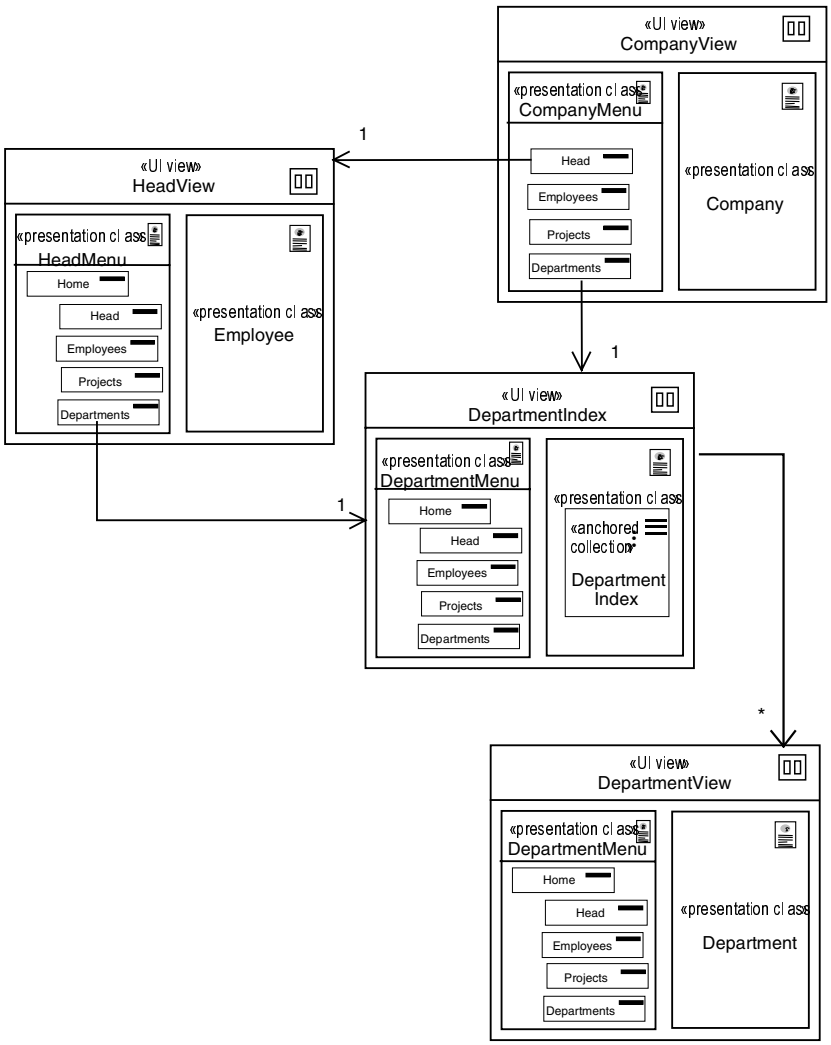

Fig. 8: Storyboard Scenario for Head and Department

4 Building the Presentation Model

The focus of this step is to decide *where* the navigation objects and access elements will be presented to the user, i.e. in which frame or window the content is displayed and which content will be replaced when a link is activated. First of all, the designer has to specify whether a single or multiple-window technique is used, whether frames are used and, if so, into how many frames framesets are divided. In the case of one window without frames the result is obvious from the storyboard model and no further graphical representation is needed. Each click produces just a complete replace of the window content by a new content. The dynamics of the presentation is optionally shown by a presentation flow.

4.1 Modeling Elements

A presentation model of a Web application is built with stereotyped classes «window», «frameset», «frame», «client page», «server page», «client script» and «server script». The user interface elements defined in Section 3, are used as well, but now in a more elaborated context. We also use the following stereotyped associations: «builds», «redirect», «submit», «link» and «displays». Many of the stereotyped modeling elements stem from the WAE of Conallen and can be smoothly integrated in our approach.
To indicate the location of a presentation we use windows and frames (which both can be targets used in an anchor). Server pages, client pages and framesets specify contents (generally called Web pages) which are displayed in a target. Those modeling elements which are new or differ semantically from the definitions in [Cj99] are listed in the following.

Target. Target is an abstract class used to generalize the concept of window and frame, both being compartments of the screen in which Web pages are displayed.

Window. A window is the area of the user interface where presentation objects are displayed. A window can be moved, maximized/minimized, resized, reduced to an icon and/or closed. For performing these actions a window contains special buttons. In addition, windows include two scrollbars: a horizontal and a vertical scrollbar that allow for visualization of the whole content of the window. Any window is an instance of a class stereotyped by «window» with a corresponding icon (see Figure 9).

Frameset. A frameset is a modeling element used to define multiple visualization areas within a window. It is divided into lower level location elements – so called frames – and may also contain an arbitrary number of nested framesets. A frameset is an instance of a frameset class stereotyped by «frameset» with a corresponding icon (see Figure 9).

Frame. A frame is always part of a frameset, it defines an area of the corresponding frameset where content is displayed. A frame is an instance of a frame class stereotyped by «frame» with a corresponding icon (see Figure 9).

Displays. The «displays» association specifies that in the target a Web page is displayed (frame or window). It is a directed association.

Links. The directed association ´links[a] points from an anchor to a Web page. Its semantic is an individual connection among two objects.

Targets. A directed association ´targets[a] points from an anchor to a target. It specifies that this target is used to display the object linked by the anchor.

Figure 9 shows a metamodel for the above modeling elements defining the well-formedness rules for their use in concrete presentation models. As already mentioned, all class names occurring in the metamodel will be used as stereotypes in concrete models.

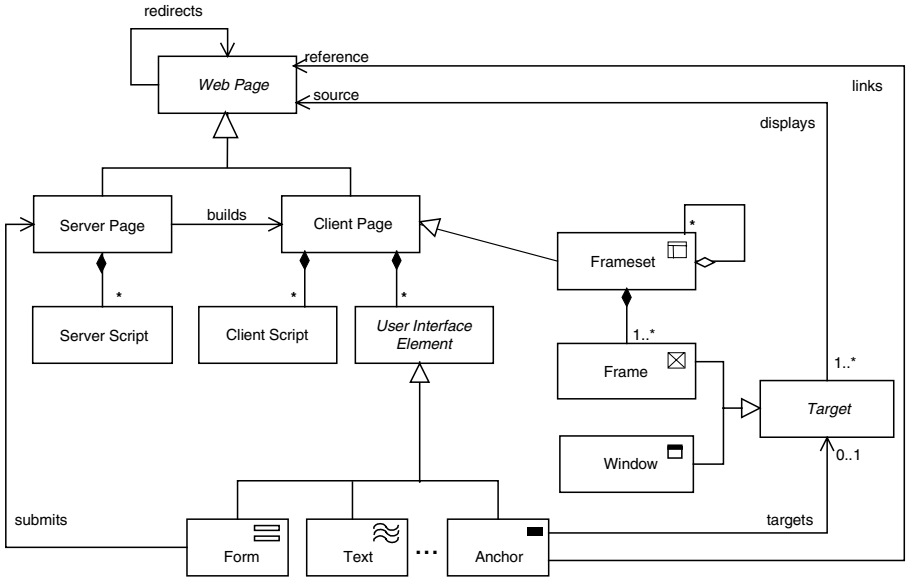

Fig. 9: Modeling Elements for the Presentation Model (Metamodel)

A presentation model must be built conform to the composition structure of classes shown in Figure 9. Consequently, these stereotypes are restrictive stereotypes in the sense of [BGJ99].

4.2 Example

Figure 10 shows (part of) the presentation model for the Web site of the service company. Figure 11 shows (using a UML sequence diagram) a presentation flow representing a scenario for a sequence of possible navigation activities that can be performed by a user of the application.

4.3 The Method

The presentation model requires that the designer takes some decisions, such as number of windows to be used and number of frames each frameset is divided into. Hence the

construction of the presentation structure cannot be fully automated, but there are certain guidelines that the designer can follow:

(1) Select between a single or multiple-window technique. In case of a multiple-window technique plan how many windows will be used.
(2) Decide whether frames will be used for the realization. If this is the case specify how many frames each frameset has. Typically the partitions of the user interface views of the abstract user interface model will be realized by frames.
(3) Transform the presentation classes of the abstract user interface model into Web pages of the presentation model. In most cases this is straightforward since only the anchors and forms have now to be directed to appropriate Web pages or server scripts by using associations stereotyped by ´link[a] or ´submits[a], respectively. If a client page should be dynamically generated a corresponding server page and an association stereotyped by ´builds[a] must be introduced.
(4) Decide in which frame (of a frameset) or in which window a Web page is to be presented to the user. This will be modeled by the target associated to an anchor.

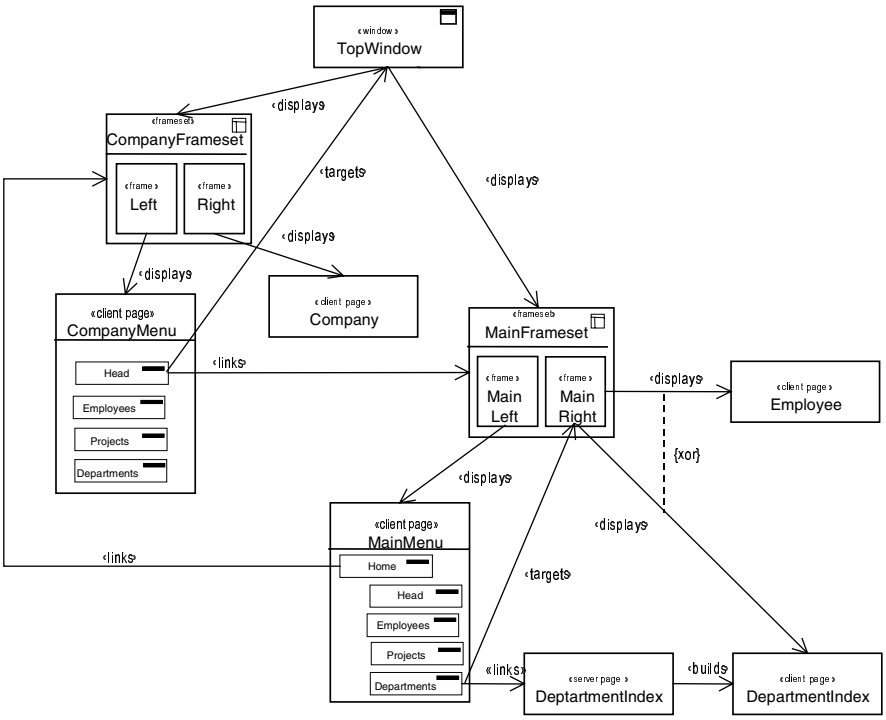

Fig. 10: View of the Presentation Model of the Company Web Site

If many windows and/or frames are used, it is advisable to construct partial views of the presentation model to avoid an overloaded notation.

The construction of the presentation flow can follow the guidelines listed below. Note that in this representation it is not relevant whether the pages are static HTML pages or they are generated dynamically.

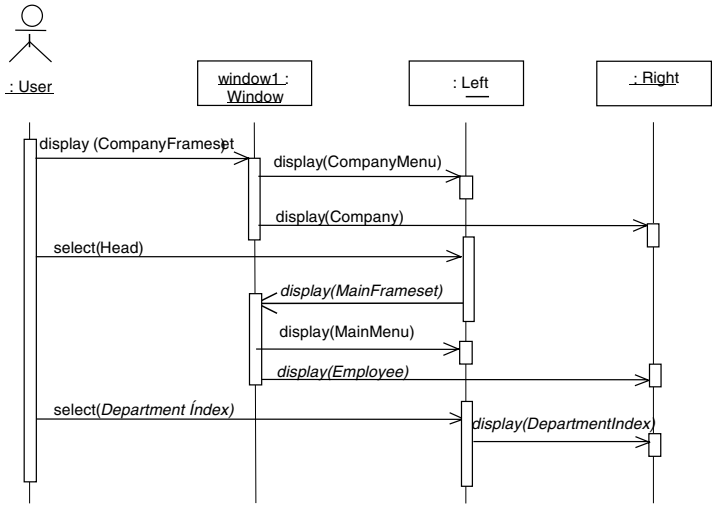

Fig. 11: Presentation Flow for a Scenario of the Company Web Site

(1) Set the scenario for the interaction model, i.e. define which navigation path of the navigation structure diagram will be modelled.
(2) Represent the user, the windows and/or frame objects in the horizontal dimension.
(3) Specify a *display* message for each presentation object that should be presented to the user (in a window or frame). The parameter of the display message is the corresponding presentation object (described in previous sections).
(4) Include a *select* message for each user action which selects an anchor or a button. The anchor or button names are the parameters of the message.
(5) Specify a *fill* and a *submit* message for each user action which consists of supplying data in a query form. This form is the parameter of the message.
(6) Include a message for each open and each close of a window.
(7) Use "balking" to specify the period of time that a window or frame is active.

5 Conclusions and Future Work

In this paper we propose the use of a UML Web Profile to model the user interface of Web applications. This approach focuses on the presentational aspects already outlined in [HK00], but there the focus was on the systematic modeling of the navigation. We present stereotypes that are added to our UML extension for Web applications and show how these stereotypes can be integrated in the design with the WAE stereotypes of Conallen.
Using the navigation model as a starting point we construct the presentation model in two steps. The first one aims to sketch the content of the nodes showing the look and feel

of user interface views of the Web application. These views are used to build storyboarding scenarios that are an excellent basis for the development of the realization of a prototype. The second step goes towards an implementation. At this point the designer has to take decisions about multiple-window technique, the use of frames and dynamic page generation. The objective of the presentation model is to show *where* the contents, after being realized in client pages, are presented to the user, i.e. in which target they are displayed and which other content of the frame or window is replaced. Additionally, server pages are identified when they are required for the dynamic construction of client pages. The methodical design of the presentation of a Web application is part of the UWE approach. The strength of our methodology is given by the fact that some steps can be performed in an automatic way thus providing the basis for a generation mechanism in the Web design.

In future work we plan to elaborate a more formal and detailed specification of the modeling elements in the metamodel supplemented with well-formedness constraints written in OCL.

An important next step is to extend existing Case Tools, such as ArgoUML [Au01], with the set of stereotypes of our Web Profile to allow for visualization of the stereotypes through the corresponding icons. In addition, a Case Tool that supports the development of Web applications based on our method is planned. It will consist of different editors for conceptual, navigation and presentation design. The objective is to systematize as much as possible the process according to the steps specified in our methodology. It will prove consistency by using the given well-formedness rules for the modeling elements.

Acknowledgements

The authors like to thank the anonymous reviewers, whose comments helped to improve the presentation of this paper.

References

[Au01] ArgoUML: The Cognitive CASE Tool. http://argouml.tigris.org (2001)

[BKM99] Baumeister, H., Koch, N., Mandel L.: Towards a UML extension for hypermedia design. In Proceedings «UML»'99, France, R., Rumpe, B. (Eds), LNCS, Vol. 1723. Springer-Verlag (1999) 614-629.

[BGJ99] Berner S., Glinz M., Joos S.: A classification of stereotypes for object-oriented modeling languages. In Proceedings «UML»'99, France, R., Rumpe, B. (Eds), LNCS, Vol. 1723. Springer-Verlag (1999) 249-264.

[Cj99] Conallen J.: Building Web Applications with UML. Addison-Wesley (1999).

[HK00] Hennicker R., Koch N.: A UML-based Methodology for Hypermedia Design. In Proceedings «UML» 2000, Evans, A., Kent, S. (Eds), LNCS, Vol. 1939. Springer-Verlag (2000) 410-424.

[JBR99] Jacobson I., Booch G., Rumbaugh J.: The Unified Software Development Process. Addison Wesley (1999).

[JFC00] Jardim Nunes N., Falcao e Cunha J. (2000). Towards a UML Profile for Interaction Design: The Wisdom Approach. In Proceedings «UML» 2000, Evans, A., Kent, S. (Eds), LNCS, Vol. 1939. Springer-Verlag (2000) 101-116.

[Ko00] Koch N.: Hypermedia Systems Development based on the Unified Process. Technical Report 0003, Ludwig-Maximilians-University Munich (2000).

[KKH01] Koch N, Kraus A., Hennicker R.: The Authoring Process of the UML-based Web Engineering Approach (Case study). Proceedings of the 1st International Workshop on Web-oriented Software Technology, 2001, to appear.

[PP00] Pinheiro da Silva P., Paton N.: UMLi: The Unified Modeling Language for Interactive Applications. In Proceedings «UML» 2000, Evans, A., Kent, S. (Eds), LNCS, Vol. 1939. Springer-Verlag (2000) 117-132.

[Pj94] Preece J., Rogers H., Benyon D., Holland S., Carey T.: Human-Computer Interaction. Addison Wesley (1994).

[Sd96] Sano D.: Lare-Scale Web Sites – A Visual Design Methodology. Wiley Computer Publishing (1996).

[Sb98] Shneiderman B. Designing the User Interface: Strategies for effective Human-Computer Interaction. Addison Wesley (1998).

[UML01] UML: The Unified Modeling Language. http://www.omg.org/uml/ (2001)