Modeling Web Applications

S Moharana*

The popularity of the Internet and the World Wide Web has resulted in the growth of the web-based systems. Not only their number has grown manifold but also they have become more complex and sophisticated in functionality. These web-based systems have become an integral part of our business and daily lives. However, development of these systems reportedly lacks systematic approach, quality control procedures and integrity as they evolve over time. This has led to the growing concern in the software development community. Several modeling approaches have been proposed to specify web applications to mitigate this concern. This article introduces the state-of-the-art developments in the modeling web applications.

Introduction

The Internet and the World Wide Web are no doubt the great developments in the history of Information Technology. Traditional websites have progressively evolved from the browsable and static information repositories to web-based distributed systems integrating databases and complex business functionalities with multimedia technologies. The popularity of web-based systems has grown manifold. Legacy information and database systems are being migrated to the Internet and web environment. These web-based systems have a significant impact on business and industry as well as on the daily lives of the people in the modern society. Electronic business is rapidly growing and making every business a global business removing its boundaries.

However, the development approach used for web-based systems has been *ad hoc* and lacks a systematic engineering process. As the complexity and sophistication of the web-based systems grow, there is a legitimate and growing concern for a systematic approach to specify and develop these systems since abstraction and modeling can manage complexity in a better way. So, several approaches have been proposed to address this concern. Currently there is no standard notation accepted industry wide for specifying the web applications. Many tool vendors adopt the UML to the web context making extensions to it; even then this adoption is not trivial.

This article is organized as follows. Section 2 discusses the nature of web applications that makes it different from the traditional software systems and shows how the existing

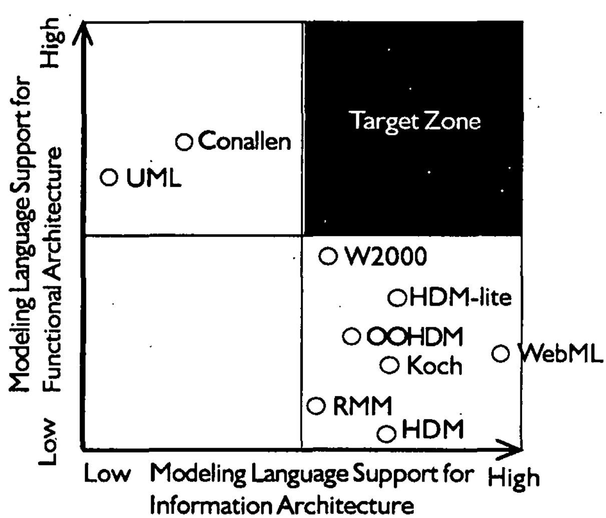
* Senior Software Engineer, TechNet Solutions, Hyderabad. E-mail: ismjolly@yahoo.co.in

modeling approaches fit in the requirements. Section 3 briefly discusses the efforts put in to extend the UML to capture different aspects of the web applications. Then it concludes.

Web Modeling Requirements

In the early days of WWW, websites were considered as hypermedia information repositories. With the popularity of the web, the requirements changed a lot. These websites have evolved to web applications having many business functions carried over them [1]. They are sophisticated and complex like any other traditional software systems in an organization. They are different from the traditional software systems in several ways like concurrency, availability, unpredictable load, evolutionary nature, security, information presentation, navigation and aesthetics [2]. To accommodate these aspects a modeling language for web applications should facilitate the requirements capture, documentation, specification, design, visualization, construction, maintenance and evolution efficiently. Since a web application is regarded as an extension of the early day websites, modeling of it should address two broad aspects. One is hypermedia modeling that takes care of information architecture and navigation of it. The other is functional modeling that captures the application functionality similar to traditional software applications. However, the existing modeling approaches fail to address both the aspects. Some like UML are good at functional modeling but poor at hypermedia modeling. Others like WebML [8] efficiently handle hypermedia modeling while giving less attention to functional modeling. Thus, there is a big gap between web application requirements and the capabilities of the existing modeling approaches. Figure 1 tries to position the various modeling approaches based on their

Figure 1: Existing Modeling Approach Gap Analysis



Source: Alice Gu, University of Technology, Sydney.

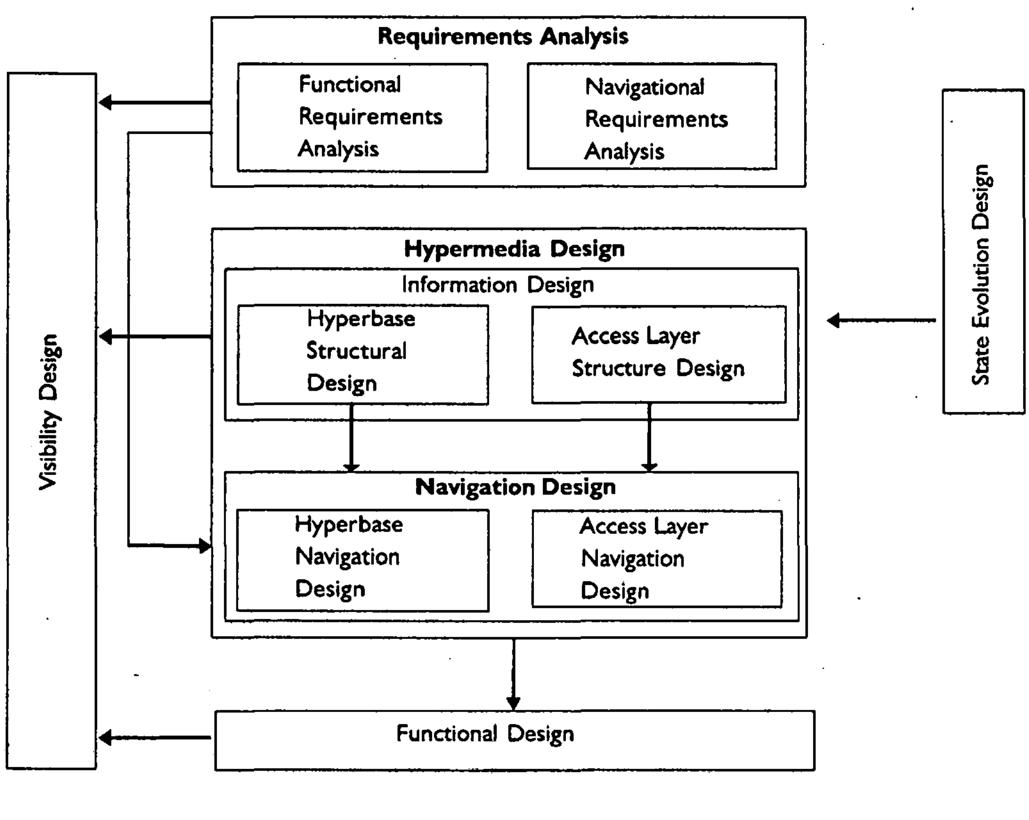
suitability for modelling web applications of today for these two broad categories of requirements [3].

Extending UML for Web

Several attempts have been made by researchers to extend the UML to accommodate the modeling requirements of a web application. Notable among them are the works by Conallen, Koch and Paolini [4], [5], [6]. However, OMG (Object Management Group) has not yet adopted these extensions to the official specification of the UML. Selecting UML as the basis for a modeling language for web has several advantages. First, it has rich expressive power with strong formal semantics. Second, it has extension mechanism through stereotypes. Third, it is process independent. Fourth, it is industry wide accepted as the standard modeling language and popular with several tools supporting it. Fifth, it has strong capabilities for functional modeling of a software system.

In their works, Paolin *et al.* propose a design framework called W2000 that blends UML and HDM (Hypermedia Design Model) for modeling web applications through

Figure 2: W2000 Design Framework



several activities [7]. Each activity produces a model with the help of diagrams that describe some aspects of the web applications. The framework is given in Figure 2.

Requirements analysis includes navigational requirement analysis in addition to the functional requirement analysis in traditional software system. The navigational requirement analysis highlights the main information and navigational structure with respect to the different users interacting with the web application. UML Use case diagram is extended for the navigational requirement analysis. State evolution design addresses the evolutionary characteristics of the web application. An information object and its navigation paths are subject to evolution. An information object may go through different states over time. This is modeled by state evolution diagram. UML state chart diagram is extended to handle this. Hypermedia design is to specify information structure and navigation path required for it keeping different user interactions. Information design uses a stereotype called entity and the semantic relations among entities. An entity type describes a class of information objects perceived by users. Components are parts of entities. They are information subunits that are not self-contained, but have well-defined roles within entities. Components may have their own subcomponents. For example, an "Article" entity in a web-based journal application has components like "Abstract", "Full-Text", "Reviews" etc.

Semantic associations denote domain specific binary relationships that exist between entities and their components. Navigation design defines navigational nodes and links of the application. Nodes are the structural elements (entities, components) of the information design. Users navigate these nodes. They are usually rendered as web pages. A link is a path that connects two nodes. Links are represented with UML arrowed associations. Navigation design is not automatically derived from information design but must be consistent with it. Figures 3 and 4 give information design and navigation design of a paper entity in a web-based journal application. The symbol "@" indicates the default node. All users who access the article must start navigation from this node. Since the "Article" entity has more than one "Reviews", it is expressed as an index with a symbol ≡. Sometimes navigation design utilizes navigation patterns and this index is an example of navigation design patterns.

Navigation design defines navigational nodes and links of the application

Functional design depicts main user operations of the application. These are found from functional Use case diagram and have scenarios. Extended UML interaction diagrams, both sequence and collaboration diagrams, are utilized for this. Visibility design deals with the specification of operations, information structure and the navigation paths visible to different users. It specifies what is available to whom.

Koch *et al.*, in their works, propose a design methodology for web applications, which consists of conceptual, navigational, and presentation designs [6]. This is based on the UML-based Web Engineering Approach (UWE) (See Figure 5).

Figure 3: Information Design

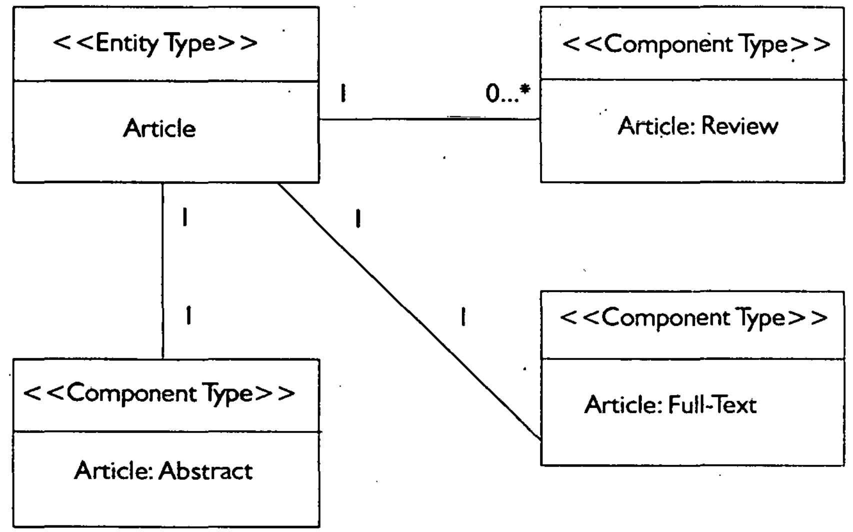
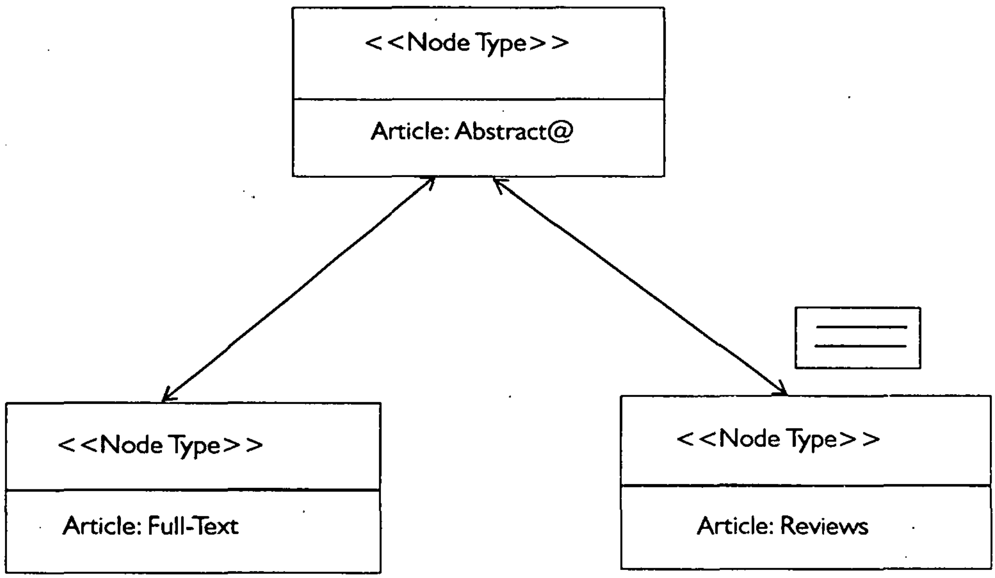
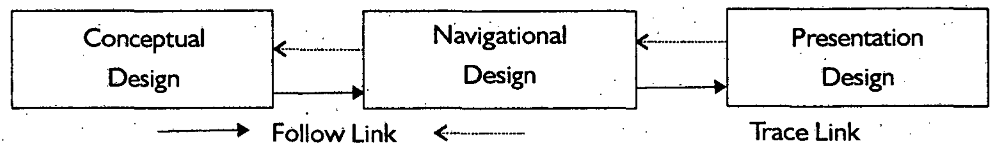


Figure 4: Navigation Design

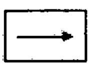
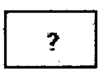
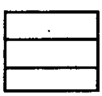


Conceptual design builds a conceptual model of the web application that takes the functional requirements captured in the Use case model. The conceptual model is represented by the UML class diagram. Navigation design is based on the conceptual model and has two components, the navigational space model and the navigational structure model. The navigational space model defines a view of the conceptual model

Figure 5: UWE Process

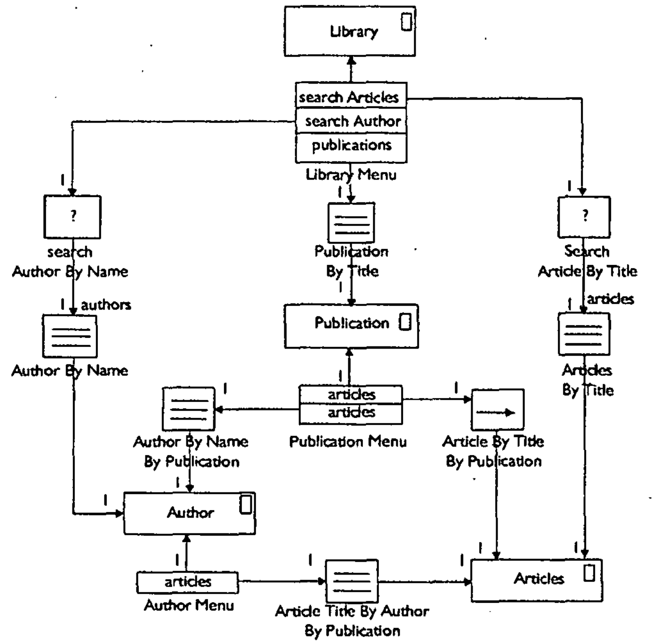


showing which classes of the conceptual model can be visited through navigation in the web application. A UML stereotype called navigation class is utilized to represent this. The navigational structure model defines the navigation of the application that shows how the navigational objects are visited. It utilizes a number of stereotyped model elements called access primitives like menus, indexes, guided tours, quires, external nodes, and navigational contexts. These stereotypes are described below.

Model Element	Description
 Navigational Class	This represents a conceptual class whose instances are visited by the user during navigation. Association in navigational model shows direction of navigation.
 Index	An index is a composite object, which contains an arbitrary number of items, and each item in turn is an object having a name and owns a link to a navigational class.
 Guided Tour	A guided tour is an object, which provides sequential access to the instances of a navigational class.
 Query	A query is represented by an object, which has a query string as an attribute.
 Menu	A menu is a composite object, which contains a fixed number of menu items. Each menu item has a constant name and owns a link either to an instance of a navigational class or to an index, guided tour or query.

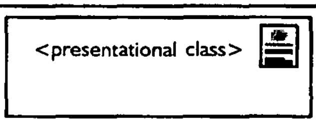
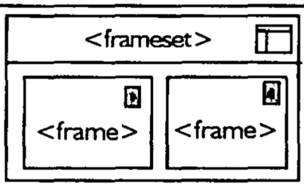
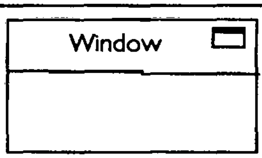
A navigation model for an Online Library application based on the above techniques is presented in Figure 6.

Figure 6: Navigational Structure Model for the Online Library



Source: [6].

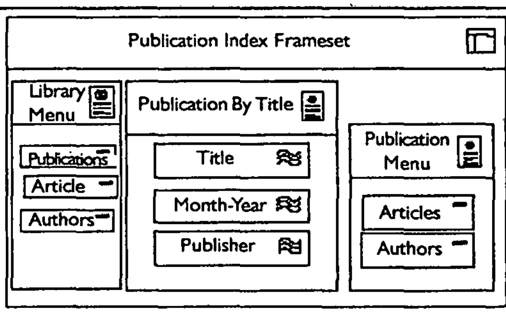
The presentation design models user interface and shows how the navigational structure is presented to the user. It utilizes a number of modelling elements given below.

Model Element	Description
 Presentational Class	It models the presentation of a navigational class or an access primitive. It contains model elements like texts, images, video, anchors etc.
 Frameset and Frame	A frameset is top-level composite element that contains lower level presentational elements. An area of the frameset is assigned to a lower level element, frame.
 Window	A window is the area of the user interface, where framesets or presentational objects are displayed.

<text> MyText Text	A sequence of characters.
<anchor> MyAnchor Anchor	A clickable text which represents a navigational link.
<image> MyImage <audio> MyAudio <video> MyVideo Image, Audio, Video	Image, Audio and video are multi media objects.
<form> MyForm Form	Use submits information using a Form that contains a number of input fields.
<collection> MyCollection Collection	A collection is a list of text elements that is used as a composite.
<anchored collection> MyAnchoredCollection Anchored Collection	A list of anchors used as a composite.

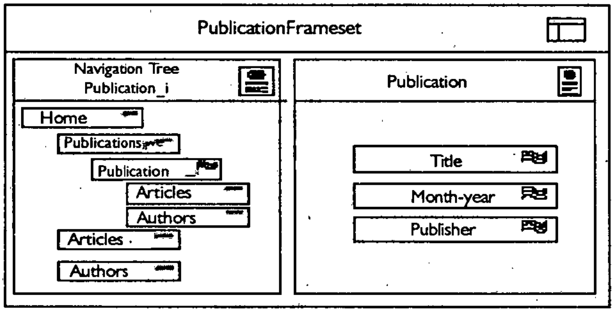
Presentation model can be different for the same navigational structure based on the restriction on the target hardware and software platforms. So, a presentation model for the Online Library can be as shown in Figures 7 and 8.

Figure 7: Menu-based Variant of the Presentation Model



Source: [6].

Figure 8: Map-based Variant of the Presentation Model



Source: [6].

Conclusion

Modeling of web applications is different from modeling other software systems. It not only has functionality but also information contents and its navigation that are equally important. The article has explored the requirements of web application and positioned different modeling efforts to fit in these requirements. It discusses some of the efforts that extend UML to model web applications. While extension to UML is a better approach than reinventing the wheel with an entirely new modeling language, it is important to note that the extension mechanism should capture all the aspects of web application efficiently. The modeling elements must have clear semantics and should integrate OCL (Object Constraint Language) for better formalism. Applying efficient techniques, the models should be derived at different stages of system development life cycle. These models should be consistent and traceable at any stage.

Although there are some issues like these with the approaches proposed, adoption of UML is justified. Future versions of UML may suitably be enhanced to accommodate the modeling requirements of web applications.

Reference # 35J-2005-12-02-01

Acknowledgment: *The author likes to thank the Consulting Editor of this journal for giving valuable insights and editing the paper.*

References

1. Baresi L, Garzotto and Paolini P, "From Websites to Web Applications: New Issues for Conceptual Modelling", Procs. ER'00, Springer, pp. 89-100.
2. Pressman R S, *Software Engineering: A Practioner's Approach*, 6th Edition, McGrawHill International, 2005.
3. Alice Gu, Brian H S, and David L, "Web Modeling Languages: The Gap Between Requirements and Current Exemplars", Procs. AusWeb 2002, The Eighth Australian World Wide Web Conference, July 2002.
4. Conallen J, *Building Web Applications with UML*, Addition Wesley, 1999.

5. Baumeister H, Koch N, and Mandel L, "Towards a UML Extension for Hypermedia Design. In proceedings UML'99 conference, France, Rumpe N (Eds), LNCS Vol. 1723, Springer-Verlag, 614-629, 1999.
6. Koch N, Baumeister H, Hennicker R and Mandel L, "Extending UML to Model Navigation and Presentation in Web Applications", Procs. Modelling Web Applications in the UML workshop, ML'2000, York, October 2000.
7. Baresi L, Garzotto G and Paolini P, "Extending UML for Modeling Web Applications", Procs. 34th Hawaii International Conference on System Science, 2001.
8. Ceri S, Fraternai P, and Bongio A, "Web Modeling Language (WebML): A Modeling language for designing Websites", Procs. of International conference WWW9/ computer Networks 33, 137-157, 2000.

Hybrid Virus

A hybrid virus (sometimes called a multi-part or multipartite virus) is one that combines characteristics of more than one type to infect both program files and system sectors. The virus may attack at either level and proceed to infect the other once it has established itself. Hybrid viruses can be very difficult to eradicate and, unless completely eradicated, will often re-infect the host system repeatedly.

In general, viruses fall into one of three classes: macro viruses, file infectors (also known as program infectors), and system or boot-record infectors. Macro viruses, which are fairly common and often less harmful than other types, infect a word processing application and typically insert unwanted words or phrases. A hybrid virus usually combines the approaches of the two latter types in order to maximize damage and resistance to removal. File infector viruses attack executable files on your hard drive. Each time you run the file, you unknowingly invoke the virus which, in turn, delivers its payload to your system. System or boot-record infectors infect executable code found in certain system areas on a disk, infecting the portion of your hard drive that contains the operating system instructions telling the computer how to start up. These viruses are invoked each time the computer starts.

Because getting rid of a hybrid virus can be such a difficult process, most security experts recommend prevention rather than cure, and suggest that people follow common sense security procedures; these include: running good anti-virus software and keeping virus definitions updated, practicing caution with e-mail and never opening an unexpected attachment or downloading a program from a questionable source.

Source: http://searchsecurity.techtarget.com