Reverse Software Engineering with UML for Web Site Maintenance

Sam Chung
Department of Science and Mathematics
College of Arts and Science
Univ. of Texas of the Permian Basin
Odessa, TX, 79765
chung_s@utpb.edu

Yun-Sik Lee
Computing Engineering
& Graduate School of Venture
Hoseo University
Asan, Chungnam 366-795 Korea
yslee@office.hoseo.ac.kr

Abstract

In this paper, we show that reverse software engineering using the Unified Process and visual models with Unified Modeling Language can be applied to web site maintenance. By reverse engineering the current web sites, the implementation models of the current web sites are derived from the web sites. For the navigation schemes, the web elements and their dependencies of the current web sites are shown in components diagrams. Also, the physical directory structures are shown in the component view of the implementation model. Our empirical results on official university web sites maintenance show that the reverse software engineering and visual models can help web administrators to understand the navigation schemes and physical structures promptly and clearly.

1. Introduction

With the advent of WWW technology, many organizations have developed their own web sites. Since anyone can create web pages and the web pages can be connected through the Internet, the size and the number of the web sites have very rapidly increased. However, the maintenance of current web sites is getting difficult due to the quick and large development of hypertext documents. This results in a lack of personnel to do the maintenance. In addition to the limited personnel resources, the contents and structures of web sites need to be changed constantly according to the new requirements of each organization. Also, the integration of new web technology such as Cascading Style Sheet, Javascript, and Java Applets with existing web sites need more thorough understanding of physical directory structures and navigation schemes from web administrators and/or developers.

In this paper, we show that reverse software engineering using the Unified Process (UP) and visual models with Unified Modeling Language (UML), which have been accepted in software industry [1, 3, and 7], can be applied to web site maintenance. To maintain a web site, the navigation schemes and the physical directory structures of the current web sites need to be known to web administrators. By reverse engineering the current web sites, the implementation models of the current web sites are derived from the web sites. For the navigation schemes, the components and their dependencies of the current web sites are shown in components diagrams. Also, the physical directory structures are shown in the component view of the implementation model. Our empirical results on a university web sites maintenance show that the reverse software engineering and visual models can help web administrators and/or developers to understand the navigation schemes and physical structures promptly and clearly.

2. Reverse Software Engineering and Visual Modeling

To maintain the current web sites promptly, the web administrators and developers are interested in the following issues:

- Navigation schemes: How are the web elements such as server page, client page, form, frame, e-mail, etc. linked?
- Physical directory structures: Where are the web pages and their directories physically located in?

However, most of information on navigation schemes and physical directory structures has been poorly maintained with limited documents. This situation is very similar to legacy software systems maintenance.

Reverse software engineering [7] is a process to transform a code into model through a mapping from a specific implementation language. How the reverse software engineering can be done depends on software development processes. There are several different

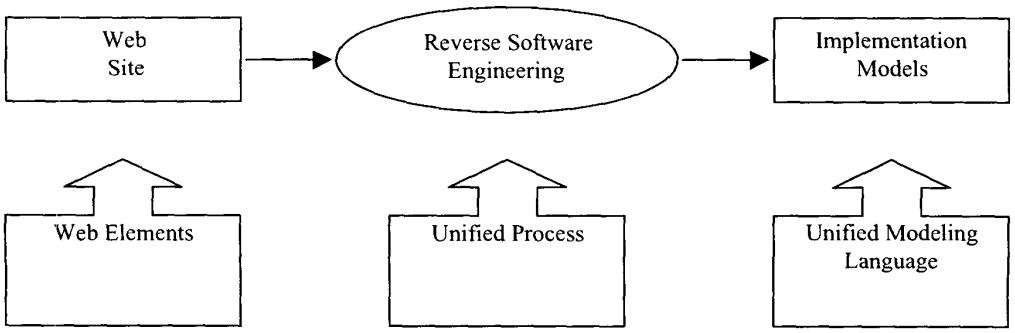
0-7695-0577-5/00 $10.00 © 2000 IEEE

software development processes: waterfall approach [7], the pinball system development life cycle [8], Rational Unified Process (UP) [7]. Especially, the UP supports 4+1 views: use-case view, design view, implementation view, process view, and deployment view. Since the UP is use-case-driven, architecture-centric, and iterative and incremental process, and its implementation view is very useful for software intensive system maintenance, it is used in this paper.

Visual modeling [1] is a process to visualize, specify, construct, and document work products in standardized diagrams. Recently, software industries have accepted UML as a standard modeling language for visual modeling [3]. UML supports 9 diagrams: use-case, class, object, component, deployment, sequence, collaboration, state chart, and activity diagrams.

Contrary to a software system in which is usually implemented by using one language, a web site is implemented with many different components in addition to HTML. However, since the web site shares many characteristics of software in terms of interface design, structure, and development process, the reverse software engineering can be applied to the web site maintenance. Also, since the visual modeling with UML can support software-intensive systems, the information for web sites maintenance such as navigation schemes and physical directory structures can be documented into the UML diagrams through visual modeling. Figure 1 shows the general concepts of reverse software engineering for web sites maintenance. From the given web site that has many different web elements such as HTML pages, images, Java applets, etc, an implementation model is derived by using UP. The implementation model consists of component view and component diagrams. To draw the diagrams, UML is used. Based upon the current implementation model, the current Web site can be maintained by Web administrators or be updated by Web developers promptly and easily.

Figure 1. Reverse Software Engineering for Web Sites Maintenance



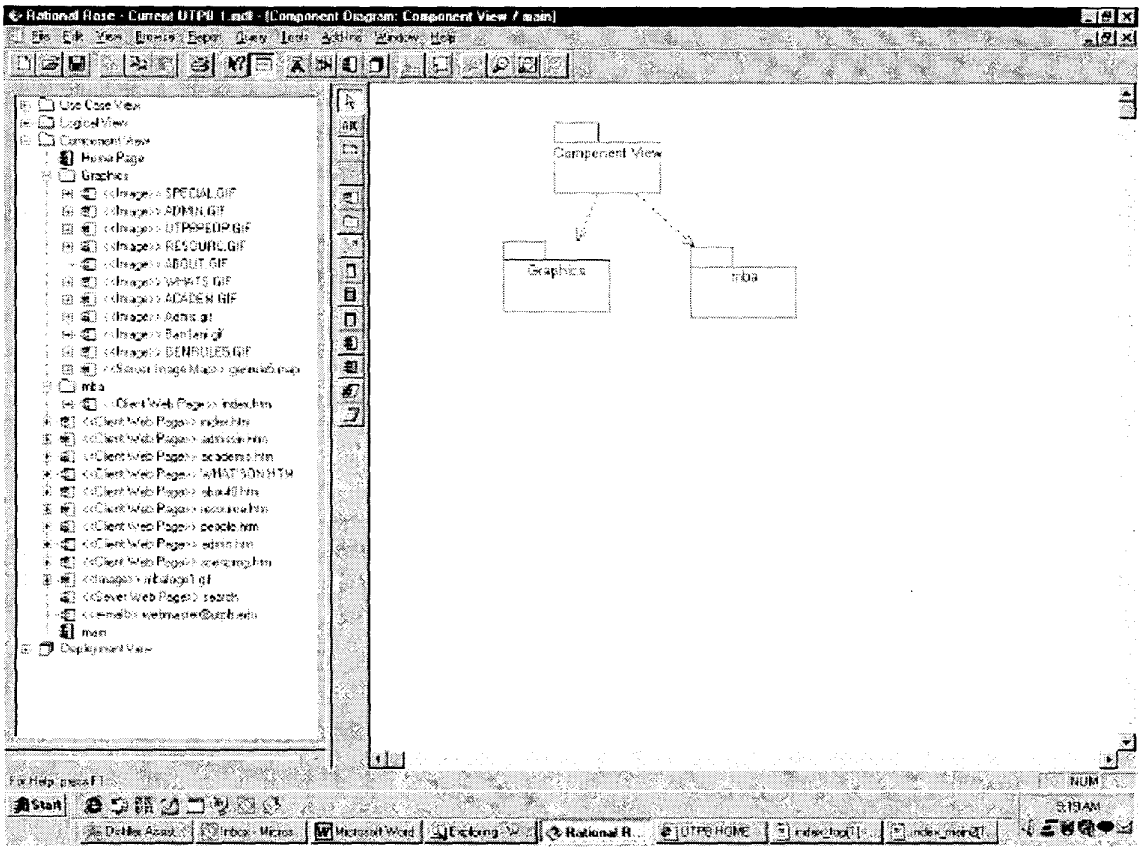
3. Modeling Web Sites

However, there is an important issue in modeling the web sites: How can we model web sites? Although many approaches for modeling hypermedia systems have been proposed in [1, 4, 5, and 6], Conallen introduced an approach for modeling web application specific elements with UML [4]. By using UML extension mechanism, Conallen showed class diagrams of the specific web elements. But, Conallen's approach is limited to design models that strongly depend upon design/logical view and class diagrams. Booch and et al in [1] used class and sequence diagrams to explain a simple Java applet. Also, they showed a component diagram for a simple web page, which consists of the applet and an image file. But, since their main concern was not to model web sites but to show an example of models with UML, the model was limited to a single web page.

In this paper, since the web administrators are much more interested in the physical directory structures and navigation schemes of their web sites, we propose a web site modeling technique using component diagrams of the current web sites and the component view of the model. The component diagrams and the component view show the model of the web site implementation. The web elements are modeled by using components and stereotypes. The navigations among the web elements are described by using dependencies among components in the component diagrams. Also, the directories and their relationships are visualized by using packages and dependency between packages in the component diagrams. The component view, which is one of 4+1

views in the UP [7], is used to show the structure of the physical components in the web sites. In the component view, a directory corresponds to a package and the physical locations of the directories are described as the hierarchical tree of packages.

Figure 2. A Component View of a University Web Site at the First Level



4. Empirical Results

As Turau mentioned in [9], the academic web sites do not have many features compared to commercial web sites. The current UTPB web site[1] has not included many important features such as diverse images, precise image maps, interactive forms, client side applications, various Internet services, cookies, etc. The university has planned better web sites to support services, academic programs, and businesses for students, faculty, staffs, and administrators. While the university web development team is renovating new web sites for the university, they have to maintain the current university web site. The maintenance and renovation of the current university web site requires the development team's understanding on the current directory structures and navigation schemes among web elements.

For explanation purpose, only the first home page of the current university web site is used in this paper. Figure 2 shows the component view of the implementation model of the web site. The home page has 1 default root directory, 2 subdirectories, and 12 web elements. The right hand side of Figure 2 shows the dependency among directories. Based upon the dependency, the navigation schemes of the first two levels of the university web site are shown in a component diagram, which is shown in Figure 3. Ea ch web element is shown using the stereotypes within << >>.

[1] The current UTPB web site: http://www.utpb.edu/indextext.htm. This web site will be replaced by the new web site, http://www.utpb.edu/index.htm, on March 1, 2000.

From the given component view and the package diagram of Figure 2, the web administrator can understand how the first level of the UTPB web site was implemented: 2 directories, 9 client web pages, 1 image file, 1 sever web page, and 1 e-mail. Since the image file and web pages are located under the same directory and there are only 2 directories, the web administrator can recognize the poor architecture of the current web site. From the give component diagram of Figure 3, the administrator can understand how components are linked: Since most of components are not dependent on the home web page, index.htm, there are one-way navigation between the top home page (index.htm) and most components. It means that a visitor can only return to the home page by clicking the back button of the web browser used. Therefore, the web administrator can recognize that navigation schemes of the first level of the web site are weak.

5. Conclusions

Web sites can be maintained more easily by using the component diagrams and the component view of the web site, which can be constructed by reverse engineering the given web site. First, since the visual models and the component view of the current web sites help the web administrators to understand the current navigation structure of the web sites and the locations and relationships of the web elements, they can maintain the current web sites much easier. Second, since the visualization of the current web sites will help web administrators and developers to have better communication for their own tasks, the quality of the web sites in navigation schemes, architecture, and contents can be improved. Third, since new visual models for the new web sites are constructed by using the current visual models and considering new requirements, the web developers/designers can renovate the current web sites with sound foundation and it will reduce the risks of the renovation project. Also, since the understanding of the current navigation schemes and the information of the web pages location and contents will help the web designer reuse many existing web pages, a lot of implementation time for the next version of the web site can be saved.

References

[1] Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The Unified Modeling Language User Guide*. Addison Wesley Longman, Inc., Reading: MA.

[2] Brereton, P., Budgen, D., and Hamilton, G. (1998). *Hypertext: The Next Maintenance Mountain*. IEEE Computer. December 1998. pp. 49-55.

[3] Coleman, D. and et al. (1998). *UML: the language of blueprints for software?* OOPSLA, 1998. pp. 201-205.

[4] Conallen, J. (1999). *Modeling Web Application Architectures with UML*. The Communication of the ACM. October 1999. Vol. 42. No. 10. pp. 63-70.

[5] Garzotte, F., Mainetti, L., and Paolini, P. (1995). *Hypermedia Design, Analysis, and Evaluation Issues*. The Communication of the ACM. August 1995. Vol. 38. No. 8. pp. 74-86.

[6] Isakowitz, T., Stohr, E., and Balasubramanian., P. (1995). *RMM: A Methodology for Structured Hypermedia Design*. The Communication of the ACM. August 1995. Vol. 38. No. 8. pp. 34-44.

[7] Jacobson, I., Booch, G., and Rumbaugh, J. (1999). *The Unified Software Development Process*. Addison Wesley Longman, Inc., Reading: MA.

[8] Ambler, S. W. (1995). The Object Primer: The Application Developer's Guide to Object-Orientation. New York: SIGS Books.

[9] Turau, V. (1998). *What Practices Are Being Adopted on the Web?* IEEE Computer. May 1998. pp. 106-108.

Figure 3. A Component Diagram of a University Web Site at the First Level