A Development Environment for Customer-Oriented Web Business

Choongseok Lee  
Korea Polytechnic University, Korea

Woojong Suh  
Inha University, Korea

Heeseok Lee  
Korea Advanced Institute of Science and Technology, Korea

INTRODUCTION

Web business can help companies strengthen the links between customers and suppliers. Under this environment, individual customers are getting much smarter and their needs are changing much faster than ever before. The customers can compare products and services with a variety of rich information provided from Web business systems, so that they can easily move to new products or services. Accordingly, many companies have conceived Web business systems to individual customers as a critical instrument for their business success and so have made a lot of efforts to develop and maintain them.

The implementation of Web business systems is complex and multidisciplinary. It requires several steps, such as information modeling for customers, navigation design for helping customers find information, user interface design for Web page layout, and actual implementation. To help develop Web business systems in a systematic fashion, many methodologies have been proposed (Lee, Suh, & Lee, 2004). The methodologies have their own advantages but are not powerful in overcoming the challenge of proper alignment of customers’ needs with the business systems. From this perspective, recently, we provided a methodology to help develop and improve the customer-oriented Web applications (Lee et al.). However, looking at the speed of change in the Web business environment, it is becoming more difficult to employ the methodology without an automated support environment. Therefore, this article presents an environment, called eBizBench, for the effective development and maintenance of customer-oriented Web business systems.

This article first discusses the previous development environments for Web business systems in a background section. Then as a main trust of this article, the architecture of the eBizBench along with its screen examples is presented, and the eBizBench is compared with other development environments. Finally, this article discusses future trends and makes a conclusion.

BACKGROUND

Currently, in constructing Web business systems developers tend to pay little attention to requirements elicitation, requirements analysis, or reusability (Taylor, McWilliam, Forsyth, & Wade, 2002). They may use ad hoc tools that depend on their expertise or experience. However, this ad hoc development of Web business systems without any rigorous development environment may cause Web crises such as delayed schedule, exceeded budget, and poor quality (Ginige & Murugesan, 2001).

To help develop Web business systems in a robust and coherent fashion, several development environments have been proposed. These development environments can strengthen the quality of the resulting Web business. They can be categorized into two types: implementation-oriented or integrated. The implementation-oriented environments focus on the generation of database and Web pages rather than conceptual design. They include WebDesigner (Bichler & Nusser, 1996), W3Objects (Ingham, Caughey, & Little, 1997), STRUDEL (Levy, Florescu, Suciu, Kang, & Fernandez, 1997), Jessica (Barta & Schranz, 1998), WebComposition Markup Language (WCML; Gaedke, Schempf, & Gellersen, 1999), ReWeb and TestWeb (Ricca & Tonella, 2001), and MyXML (Kirda, Jazayeri, & Kerer, 2001). The integrated environments cover phases ranging from conceptual design to implementation details. OOHDM-Web (Schwabe, Pontes, & Moura, 1999), Araneus (Mecca, Merialdo, Atzeni, & Crescenzi, 1999), AutoWeb (Fraternali & Paolini, 2000), JWeb (Garzotto, Paolini, & Baresi, 2001), and OO-H CASE Tool (Gómez & Cachero, 2001) belong to this integrated category. The
implementation-oriented development environments tend to evolve into integrated ones.

On the other hand, in the Web business environment, the speed of change is inconceivable; companies should be able to analyze customers’ experiences and respond to their needs agilely. For obtaining these capabilities, Web business systems also need to be developed and evolved rapidly. Yet, the previous system-development environment for Web business systems falls back on technical details such as conceptual design and visual code generation. Especially, they offer little opportunity for the kind of customer analysis over time. Accordingly, they have a limit on effectively fitting technical details into the real lives of the customers. Furthermore, the development environments can leap in with more reuses of analysis results. Getting work done nowadays depends on getting designers to share their design results. For the above customer orientation and reusability, the eBizBench was developed as an integrated development environment.

**eBizBench ARCHITECTURE**

The eBizBench supports whole activities for development in a seamless fashion, ranging from customer needs analysis to system implementation. As the development of Web business systems becomes complicated, it becomes almost impossible to build the system manually; it is important to catapult the Internet business to high quality when development time and money are scare. An automated environment for supporting development activities is desired. The proposed eBizBench employs a repository as the basis of the reuse and conversion of design results.

The eBizBench has been implemented using Microsoft Visual Basic 6.0, Active Server Page (ASP), and Microsoft SQL Server 7.0 database management system (DBMS) in the Windows environment. The DBMS manages the repository. The overall architecture of the eBizBench is depicted in Figure 1. The eBizBench consists of seven subsystems: (i) project management, (ii) customer analysis, (iii) value analysis, (iv) Web design, (v) implementation design, (vi) repository management, and (vii) repository.

The project management subsystem interacts with the other subsystems. It can manage a variety of system development projects.

The customer analysis subsystem consists of the customer need analysis supporter and customer correspondence analyzer. Customers are categorized into customer groups according to their features. Each customer group has different needs. Customer groups and the corresponding needs are summarized in the customer need analysis supporter, and then their correspondence is analyzed by the use of the customer correspondence analyzer, as shown in Figure 2.

The objective of the value analysis is to model value activities. This subsystem consists of the value derivation designer and value activity modeler. In the value derivation designer, customers’ needs are analyzed and then prioritized in the form of the value derivation table. The value activity modeler identifies customers’ requests in the form of events, as shown in Figure 3.

The Web design subsystem deals with functionalities for the conceptual design. It includes the scenario designer, object modeler, view and navigation designer, and page specification designer. The scenario designer...