A System for Developing Customer-oriented Internet Business: eBizBench

Choongseok Lee [a], Byounggu Choi [b], and Heeseok Lee [a]

[a] *Department of Management Information Systems, Graduate School of Management, KAIST*
207-43, Cheongryang, Seoul, SOUTH KOREA 130-012
Tel: +82-2-958-3654, Fax: +82-2-958-3604, E-mail: freeman@kgsm.kaist.ac.kr; hsl@kgsm.kaist.ac.kr

[b] *Carlson School of Management, University of Minnesota*
321 19th Avenue South, Minneapolis, U.S.A, MN 55455
Tel: +1-612-624-1303, Fax: +1-612-626-1316, E-mail: bchoi1@csom.umn.edu

Abstract

The Internet is a global network of networks enabling computers of all kinds to communicate and share services directly and transparently. It can lead to a completely new electronic economy. An important challenge in the age of Internet business is the proper alignment of customers' needs with the business system. To improve an Internet business system according to customers' needs continuously, reuse of design results is also of particular importance. This paper proposes a system, called eBizBench, which supports this alignment and reusability. The eBizBench consists of seven parts: (1) project management, (2) customer analysis, (3) value analysis, (4) web design, (5) implementation design, (6) repository system, and (7) repository. The eBizBench can help develop Internet business systems in a systematic fashion. The repository and the repository subsystem are useful not only for the reuse and conversion of design results among design support systems, but also for the basis of maintenance of Internet business system. A real-life case is illustrated to demonstrate the feasibility of the eBizBench.

Keywords:

Internet Business System, Customer-Oriented, and Support System

Introduction

The Internet is a global network of networks enabling computers of all kinds to directly and transparently communicate and share services. It can lead to a completely new electronic economy. Internet business can sharpen business operations to improve internal and external business processes. Internet business can enable commercial exchanges that cross physical, temporal, cultural, and legal boundaries on a scale that was technically infeasible [36]. Internet business can help companies strengthen the links between customers and suppliers [1, 31, 19]. Many companies have attempted to develop Internet business systems to create innovative virtual companies, markets, and trading communities [27; 37, 38]. Successful players in the new electronic economy leverage Internet technologies in every aspect of their business operations [2, 10].

Internet business systems are implemented by the use of web technologies. This implementation is complex and multidisciplinary [27]. It requires several steps such as information modeling for customers, navigation design to help customers find information, user interface design for web page layout, and actual implementation. Currently, most developers pay little attention to requirements elicitation, requirements analysis, development methodologies, and process [8, 71]. They make use of ad hoc approaches that depend on their expertise and experience. The ad hoc development of Internet business system without any rigorous or systematic approach causes Web crisis [27]. Also, most Internet business systems are implemented from system-oriented perspectives [5]. Because navigation structures are designed for the ease of implementation [40] and information is modeled for database efficiency, the resulting system-oriented Internet business systems are likely to be difficult for customers to utilize.

Internet business development methodologies have been proposed by various researchers such as Garzotto et al. [23], Lange [43], Isakowitz et al.[34], Schwabe & Rossi [60], Artz [3], Bichler & Nusser [11], Takahashi & Liang [68], Atzeni et al.[4], Troyer & Leune [72], Abels et al.[5,6], Schwabe et al. [60], Lee et al. [44, 45], Fraternali & Paolini [21], Lee & Suh [47], and Chen & Heath [15]. These methodologies focused primarily on analyzing information and relationships in data model. They fail to incorporate customers' requirements in detail. Because customers' needs change dramatically, Internet businesses evolve continuously. The development of a web site reflecting customers' needs is critical for sustaining competitive advantages [53, 66]. A customer-oriented methodology to develop an Internet business system is required to support these activities. Internet business development methodology should cover design activities ranging from customers' need analysis to system implementation. It helps developers capture customers' needs at early stage.

A critical success factor for wider use of design methodologies is the availability of software tools [39, 25]. Furthermore, no automated support exists for anything more than basic HTML editing, and barely any systems

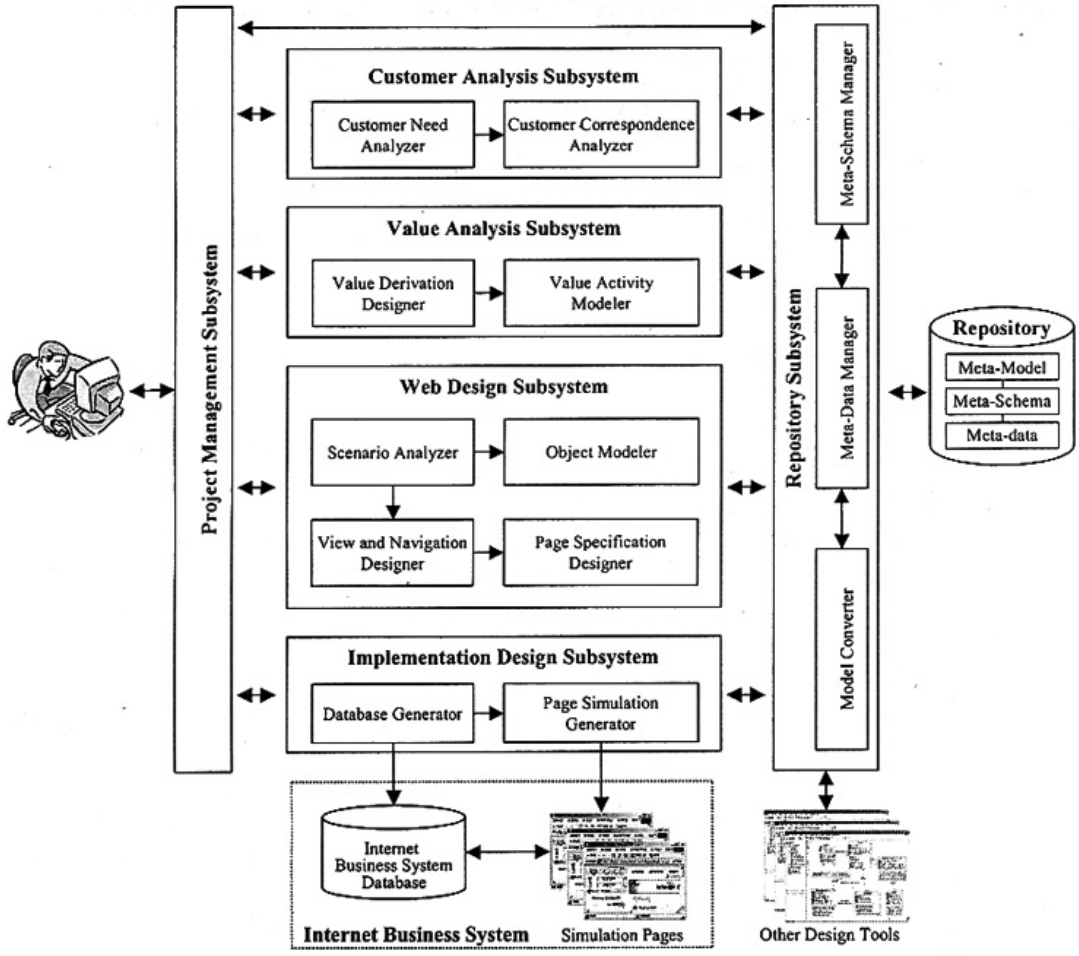

Figure 1- Architecture of the eBizBench

support advanced Internet business system development. Also design tools aren't integrated properly with the development process [8]. To support design and development of Internet business, several tools have been proposed. These tools can be categorized into two types: implementation-oriented and integrated perspectives. The former focused on generation of database and web page rather than conceptual design for Internet business system development. WebDesigner [11], W3Objects [33], STRUDEL [49], HTML++ [59], Jessica [9], WebComposition Markup Language (WCML) [22], ReWeb and TestWeb [56, 57], and MyXML [41] are implementation-oriented. The latter covered from conceptual design to implementation for Internet business system according to a model. OOHDM-Web [61], Araneus [51, 52], AutoWeb [21], JWeb [25], and OO-H CASE Tool [28] belong to integrated support systems. The implementation-oriented support systems evolved into integrated ones.

Most of these systems focused on implementation and visual code generations. There is critical need for support system that bridge not only the gap between design and implementation but also the gap between customers' needs and design. Also, only one design support system is not enough to develop different kinds of applications with different requirements [16]. It is necessary to the reuse and conversion of design results into another design support system.

This paper proposes a system, called eBizBench, which systematically supports for developing customer-oriented Internet business system by aligning customers' needs with implementation details. For this alignment, scenarios are analyzed in the eBizBench. And repository and repository subsystem are implemented not only for the reuse of design results, but also design conversion among other design support systems. The outline of the paper is as follows: The next section provides the architecture of the eBizBench. Section 3 describes the eBizBench in further detail using a real-life case. Section 4 describes a repository and repository subsystem for the eBizBench. Section 5 compares the eBizBench with others. The paper concludes in Section 6.

eBizBench Architecture

The eBizBench has been implemented to support design and development of Internet business. Internet business can be analyzed and designed manually, but only in simple and small Internet business system domain. As the range of domain is extended and the complexity is increased, however, it becomes almost impossible to analysis the domain and design Internet business system by hand. A

semi-automated system for supporting these activities is desired. This is the why the eBizBench has been implemented. The eBizBench enhances the applicability of systematic design and implement of Internet business system. It uses a repository [73] as basis of the reuse and conversion of design results.

The eBizBench has been implemented using the Microsoft Visual Basic 6.0, Active Server Page (ASP), and Microsoft SQL Server 7.0 database management system (DBMS) in Windows environment. The DBMS has been used for management and maintenance of a repository.

The overall architecture of the eBizBench is depicted in Figure 1. The eBizBench consists of seven parts: (1) project management, (2) customer analysis, (3) value analysis, (4) web design, (5) implementation design, (6) repository subsystem, and (7) repository. The functions of each module are summarized in Table 1.

The project management subsystem interacts with customer analysis subsystem, value analysis subsystem, web design subsystem, implementation subsystem, and repository subsystem. Its main role is the management of Internet business system development project. It facilitates that the eBizBench can be used in many Internet business systems projects.

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

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

The web design subsystem provides functionalities for logical design of Internet business to be implemented. Scenario analyzer, object modeler, view and navigation designer, and page specification designer are modules in the web design subsystem. The scenario analyzer describes scenario in the form of natural language. These scenarios lead to object model in the object modeler. In the view and navigation designer, information units that customers want to find are designed as object-oriented views, and then navigational paths for customers to access information conveniently are designed. Finally, through page specification designer, users' information windows and the flow from one page to another are defined according to the views and their navigational paths.

Table 1- Summary of eBizBench Functions for each module

Subsystem	Module	Function
▪ Project Management Subsystem	▪ Project Manager	▪ Retrieve previous Internet business system projects ▪ Update a previous Internet business system project ▪ Create a new Internet business system project
▪ Customer Analysis Subsystem	▪ Customer Need Analyzer	▪ Summarize customer groups and their needs
	▪ Customer Correspondence Analyzer	▪ Analyze correspondences among customer groups and their needs
▪ Value Analysis Subsystem	▪ Value Derivation Designer	▪ Identify customers' values of their needs ▪ Identify value activities to achieve customers' values ▪ Define implementation priority for customer need
	▪ Value Activity Modeler	▪ Specify value activity between customer group and subsystem
▪ Web Design Subsystem	▪ Scenario Analyzer	▪ Describe scenario for each event
	▪ Object Modeler	▪ Design objects and relationships among objects
	▪ View and Navigation Designer	▪ Define object-oriented (OO) views ▪ Define access structure nodes (ASNs) ▪ Specify navigational links among OO views and ASNs.
	▪ Page Specification Designer	▪ Define specifications for page structures. ▪ Define page navigation links among page.
▪ Implementation Design Subsystem	▪ Database Generator	▪ Generate relational database
	▪ Page Simulation Generator	▪ Generate simulation page
▪ Repository Subsystem	▪ Meta-Schema Manager	▪ Manage instances of meta-model (meta-schema)
	▪ Meta-Data Manager	▪ Manage instances of meta-schema (meta-data) ▪ Browse meta-data for specific project
	▪ Model Converter	▪ Convert a model instance into another
▪ Repository	▪ Meta-Model	▪ Store meta-information about model
	▪ Meta-Schema	▪ Store meta-information about components of a specific model
	▪ Meta-Data	▪ Store meta-information about a specific Internet business system.

The implementation design subsystem consists of database generator and page simulation generator. The object model and object-oriented views are transformed into logical database schema and then physical database are generated for target database management system (DBMS) by using database generator. The page simulation generator provides simulation of navigations among pages that specified by page specification designer.

By using a repository, the repository subsystem not only manages and retrieves the design results of each subsystem, but also converts design results of other design support tools. It facilitates that alignments and reuse of design results can be archived effectively. Meta-schema manager, meta-data manager, and model converter are modules of the repository subsystem. The repository consists of 3 layers: meta-model, meta-schema, and meta-data. The meta-data is the results of design for specific Internet business system.

eBizBench Details

In this section, each subsystem of the eBizBench is described in further detail by the use of a real-life case. This Internet business site will be referred to as " A company."

This company is one of the major companies in the infant clothes/goods industry in South Korea. The firm has the best position in the market with a 34% in the specialty brand market with competition from 20 small companies. It possesses a competitive edge, including the use of differentiated marketing strategies for each segment market/distribution channel and external procurement through the 340 companies.

Project Management Subsystem

The eBizBench starts by creating a new Internet business system project or retrieving one with project management subsystem. The project management subsystem interacts with other subsystems in the eBizBench for management of Internet business system development project. It facilitates that the eBizBench can be used in many Internet business systems projects using "New Project" and "Open Project" menus.

Customer Analysis Subsystem

Customer analysis subsystem includes customer need analyzer and correspondence analyzer. Customers are categorized into customer groups according to their features. Customer groups and the corresponding needs are summarized in customer need analyzer and then their correspondence is analyzed via correspondence analyzer.

Customer Need Analysis

In the customer need analyzer, customers are analyzed on the basis of customers' data. Primary data can be collected by observation, survey, and experimentation. Customers may be categorized according to their geographic, demographic, psychographics, or behavioral variables [42]. It is believed that behavioral variables are effective for this classification [31].

In case of the A company, customers are categorized into four groups according to their baby's age: before birth, before one-hundredth day, before first birthday, and after first birthday.

For the customer need analysis, survey for customers' needs is conducted from August 15 2002 to August 18 2002. As a result, 11,639 responses are analyzed. Customers' needs are grouped into 9 categories, as shown in Figure 2. For example, news clipping is most frequently suggested. It provides news that related with childcare for customers

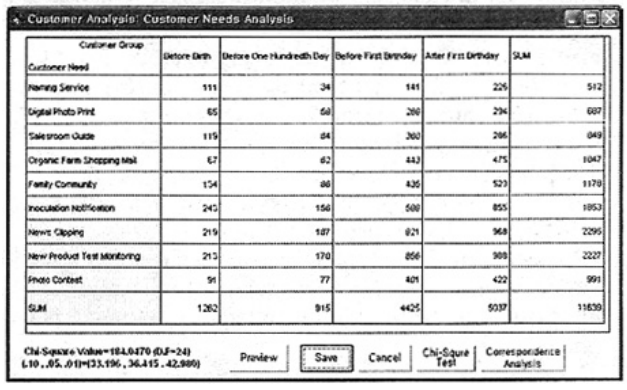

Figure 2-Customer Need Analyzer

Customer Correspondence Analyzer

The purpose of the customer correspondence analyzer is to find the correspondence between customer groups and their needs. A correspondence analysis technique [30] is adopted for this purpose. The correspondence analysis provides a mathematical method for representing data in a Euclidian space so that the results can be visually examined for structure. It is exploratory in order to analyze simple two-way or multi-way tables containing some measure of correspondence between the rows and columns.

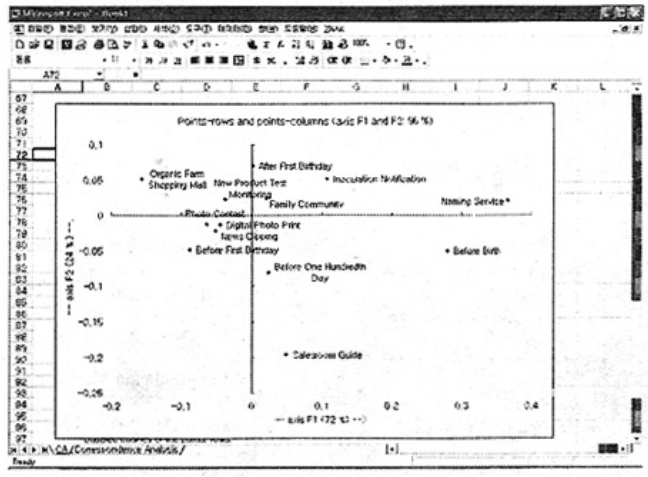

Figure 3-Customer Correspondence Analyzer

The analysis results in the chi-square value, 184.0 (degree of freedom=24). This value confirms the correspondence. Figure 3 shows the customer correspondence diagram. In

this diagram, the horizontal axis and vertical axis describe row and column coordinates, respectively. For example, the needs for customer with after first birthday child include family community, inoculation notification, and new product test monitoring. These correspondences are determined as shown in Figure 4.

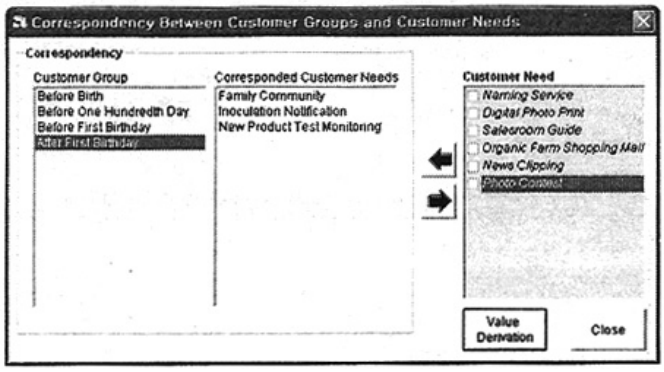

Figure 4-Determination of Correspondence

Value Analysis Subsystem

The objective of the value analysis subsystem is to identify the value of customers' needs and then describe the activities between customers and web system. It consists of value derivation designer and value activity modeler.

Value Derivation designer

Customer values can be explored on the basis of customers' activities with the system. The value derivation designer employs a value analysis table to analyze customers' values, value activities, and implementation priorities. Customer value is defined as bundle of benefits customer expects from a given product or service [42]. Customer value can be measured from financial, perceptual, and behavioral perspectives [72]. Value activity is a customers' distinct event or integrated set of events to create customer value. Because customer can't understand new opportunities made possible with new technologies or environments, implementation priorities are decided on the basis of technical feasibilities, business capabilities and strategic importance.

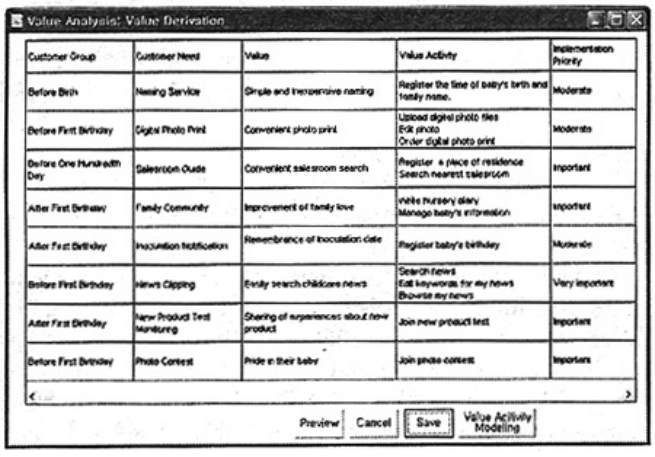

Figure 5-Value Derivation Designer

Detailed evaluation measures of the implementation priorities are outside of the scope of this paper. However, Quality Function Deployment (QFD) can be applied to evaluate implementation priorities. QFD provides capabilities how customers' needs are being translated to technical requirements and how to evaluate priorities [29].

For example, customer bring up baby whose the age of the moon is less than first birthday, highly value "Easily search childcare news" in the news clipping from customers' perceptual perspective. "Search for news," "Edit keywords for my news," and "Browse my news" are activities for achieving this value. The implementation priorities of the news clipping and the new product test monitoring are fairly high according to A Company's current capabilities and strategic importance.

Value Activity Modeling

For the sustained customer loyalty, it is important to investigate the activities between customers and the system to be implemented. At the same time, the scope of the system is determined. A value activity diagram can capture these value activities in detail. The diagram is based on events, customer groups, and subsystems. Events are identified for each customer group. An event is a trigger that starts the system [12].

In the value activity diagram, the needs for each customer group are depicted in the form of the corresponding subsystem. Each customer group triggers events for this subsystem via value activities.

Figure 6 shows the value activity diagram in our case. The emphasis is on one item for customer needs with top implementation priority ("News clipping"). This item is depicted as subsystems. The "Search news," "Edit keywords for my news," and "Browse news" activities in the value table are subdivided into 5 events: "Search news," "Insert keywords for my news," "Update keywords for my news," "Delete keywords for my news," and "Browse my news."

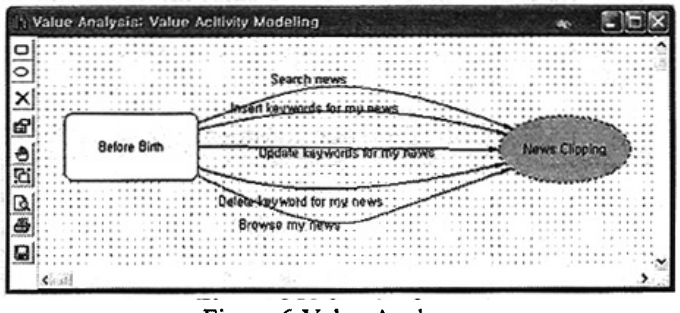

Figure 6-Value Analyzer

Web Design Subsystem

The web design subsystem consists of four modules: scenario analyzer, object modeler, view and navigation designer, and page specification designer.

Scenario Design

The eBizBench employs scenarios to identify business system requirements from the earliest opportunity by the use of natural language. A scenario is similar to a use case or scripts [35], but it is different in several ways. A use case describes interactions at a technical level. However, a scenario is a description of customers' interactions with a subsystem from the customers' perspective. Scenarios correspond to key business processes and thus can capture customers' requirements in a natural fashion.

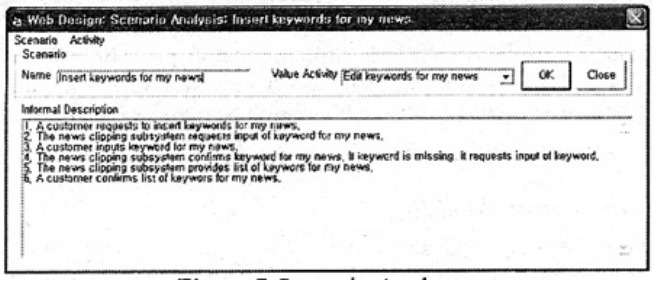

Figure 7-Scenario Analyzer

In the scenario analyzer, the events drawn from the value activity analyzer are described in the form of natural language. The natural language can enhance ease of use and understandability [58].

In the A company case, five scenarios are generated from the five events. For example, Figure 7 shows "Insert keywords for my news."

Object Modeling

The Internet business system provides a variety of information for customers. This information should be modeled. An object-oriented modeling technique has several advantages (comprehensiveness, understandability, changeability, adaptability, and reusability) compared with other techniques [32, 12]. The eBizBench adopts the object-oriented modeling technique to inherit its advantages. Objects are extracted from scenarios.

An object includes its name, attributes, and responsibilities. Attributes contain properties of the object. Responsibilities are behavioral properties of the object [75, 12].

From five scenarios, four objects are generated. The resulting object model is depicted as shown in Figure 8.

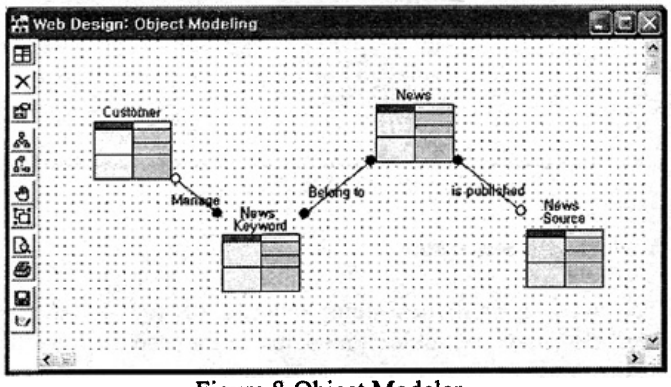

Figure 8-Object Modeler

View and Navigation Design

In the view and navigation designer, information units that customers want to find are designed as object-oriented views, and then navigational paths for customers to access information conveniently are designed.

A view is a subset of the object model; i.e., the information unit that customers want to find. These views may be referred to as Object-Oriented (OO) views. These OO views are designed on the basis of responsibilities, attributes, and relationships in the object model. Views are categorized into three types: base view, association view, and collaboration view.

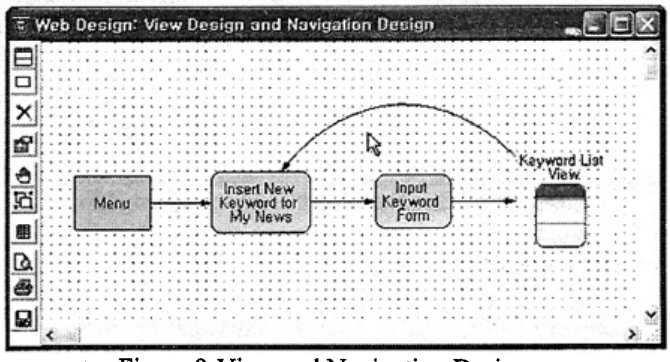

Figure 9-View and Navigation Designer

Important components in an Internet business system are nodes and links [50]. The OO view and access structure node (ASN) are adopted for navigational units. The ASN differs from the OO view in that ASN contains access paths to OO views. The OO view contains actual information that users want to obtain. These OO views and ASNs correspond to nodes. A link denotes the relationship between the source node and destination node. These source and destination nodes may be OO views or ASNs. Web pages are implemented on the basis of these OO views and ASNs in the subsequent page specification designer.

From the scenarios, ASNs are found and then navigational links are determined. First, an event that begins with a scenario can become an event ASN. An event ASN starts navigation. Second, a selection ASN can be generated when a customer should select the next activities. Third, an input ASN can be generated when a customer needs to input contents.

In Figure 9, the view and navigation designer are depicted. Three ASNs and one view are designed from "Insert keywords for my news" scenario in Figure 7.

Page Specification Designer

In the page specification designer, web pages are specified. A page is a window having information and navigational guide.

Web pages should be designed for customers to obtain necessary information easily. Each customer may require different designs. Therefore, more than one page specification may be possible; a page specification may be composed of many views and ASNs.

A page is designed according to the following steps: First, a view can be a page. Second, an input ASN and a selection

ASN can be an individual page. Third, when an ASN follows a view, a page can include both this ASN and the following view. Fourth, an event ASN can be used with an anchor in other pages.
Pages are specified by organized anchors, OO views, and additional description details (e.g., embedded components, text, images, sounds, etc.).
Figure 10 shows the screens of the page specification designer.

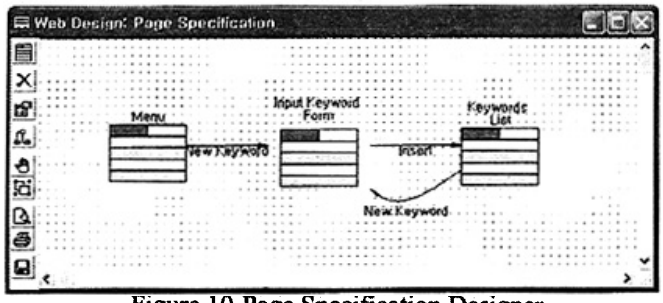

Figure 10-Page Specification Designer

Implementation Design Subsystem

The implementation design subsystem consists of database generator and page simulation generator. The object model and object-oriented views are transformed into logical database schema and then physical database are generated for target database management system (DBMS) by using database generator. The page simulation generator provides simulation of navigations among pages that specified by page specification designer.

Database Generator

The eBizBench adopts the object model for data modeling. If OODBMS is used, this object model can be transformed directly. However, in many real-life cases, RDBMSs (Relational DBMSs) are most popular. Therefore, the transformation of the object model into relational schema is required. Here, designers need guidelines for this transformation. These transformation guidelines can be found in [14, 20, 45].
In addition to these guidelines, more views, SQL (Structured Query Language) stored procedures, or SQL triggers are required especially for the transformation of the collaboration relationship. A SQL stored procedure is a user-defined program module that is stored at the database server and can be invoked by client applications. A SQL trigger is usually in the form of SQL stored procedure and is automatically invoked by the data-related event. It is typically used to perform tasks related to changes in a table [55]. By using these rules, physical database is generated automatically from object model via the database generator.

Page Simulation Generator

The page simulation generator provides simulation of navigations among pages that specified by page specification designer. The usability of Internet business system to be constructed can be evaluated using simulation.

A Repository and Repository Subsystem

The eBizBench is implemented on the basis of a repository. We propose a repository and a repository subsystem to improve two aspects as follows. First, to improve an Internet business system according to customers' needs continuously, reuse of documents is of particular importance [52]. Second, there is no existing full-featured Internet business methodologies and design support system that one can use to develop different kinds of applications with different requirements. Thus if one wants to develop deferent applications, one might need to use different methodologies and design support systems [16]. For this reason, the conversion of design results into other design support systems is very important.

A Repository

To design a repository, we adapt and slightly modify the three-layered schema of enterprise model repository (EMR), which was proposed for model independence and model integration in the schematic way [46].
Our repository schema is depicted in Figure 11 on the basis of Entity Relationship Diagram (ERD). The repository schema consists of 3 layers such as meta-model, meta-schema, and meta-data. Entities of every layer are existentially dependent on the upper layers. The meta-model provides meta-information about design models such as OOHDM, RMM, HDM, and SOHDM [45] and their relationships. The meta-schema is designed to store the design results based on design models. The meta-data is the instances of the meta-schema. For example, in a conference paper review system, a specific paper is a data. This specific paper is one of the instances of "paper" object. This paper object is meta-data and is one of the instances of object that described in the meta-schema. This object is one of the components of SOHDM design model is specified in the meta-model.

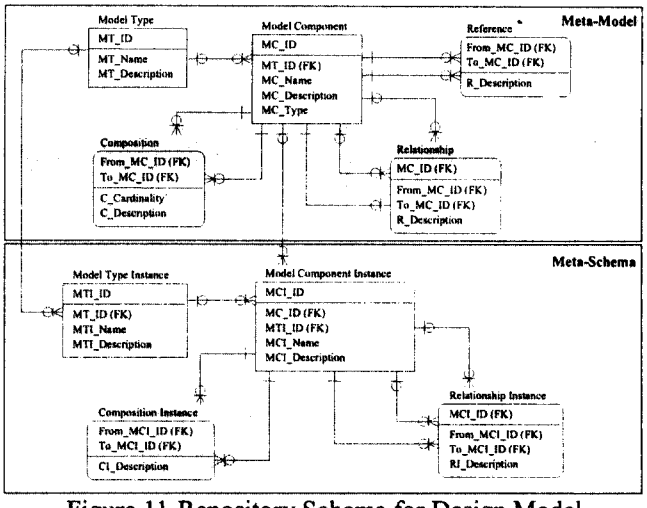

Figure 11-Repository Schema for Design Model

In this repository schema, the "Model Type" means design model for specific or general domain. OOHDM, HDM, ER,

and UML are the examples of the model types. The model type consists of several model components. The model components can be categorized into three types such as unit component, composition component, and relationship component. The unit component is independent and separated form in a model. The composition component is a constituent component of unit component or other composition component. The relationship component describes the relationship among components. The semantic similarity among model components of different models is described as "Reference." For example, navigation node and navigation link are the examples of model components of the HDM model type. Here, navigation node is the unit component and navigational link the relationship component. The node data as structural element is the composition component.

The meta-model and meta-schema can provide designers with the functionality of design conversion among design support systems in system independent way. Design results can be seamlessly aligned and can be reused using the meta-schema and meta-data.

The meta-schema is designed for model integration. It is not proper for the eBizBench to use the meta-schema directly for system performance and efficiency.

A Repository Subsystem

The repository subsystem consists of three modules such as meta-schema manager, meta-data manager, and model converter. It not only manages and retrieves the design results of each subsystem, but also converts design results of specific design model into another.

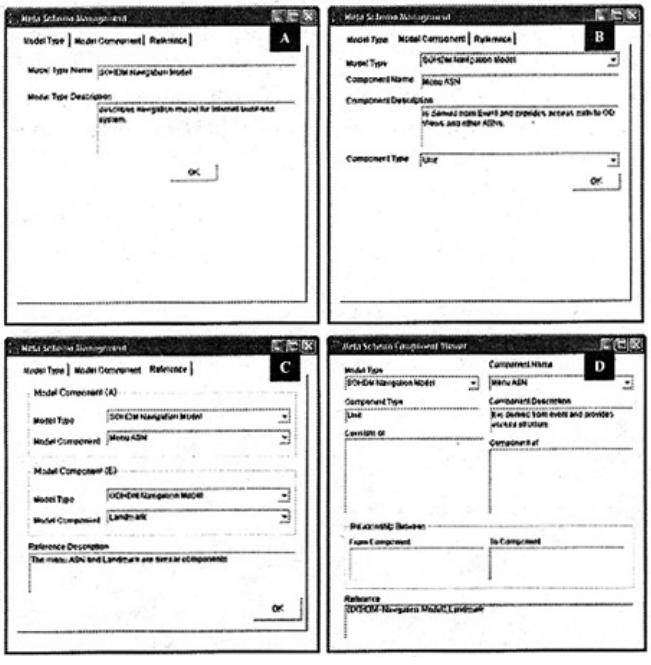

Figure 12-Meta-Schema Manager

Meta-Schema Manager

The meta-schema manger manages instances of the meta-model in the proposed repository. The meta-schema manager can add new Internet business design model and also can modify existing one according to evolution of Internet business design model in a semantic way.

Figure 12 depicts screens of meta-schema manager. In the screen A, a model can be registered. The components of the model can be specified in the screen B. The references among components of different models are specified in the screen C. In the screen D, Registered model components can be retrieved in detail. For example, the "Menu ASN" of navigation model in SOHDM and "Landmark" in OOHDM are similar and can be interchangeable.

Figure 13 shows the screen of the meta-data browser. This screen provides relationships and linkages among design results. For examples, information about "Insert keywords for my news" scenario is provided with other design results (object list, view list, and ASN list) derived from this scenario.

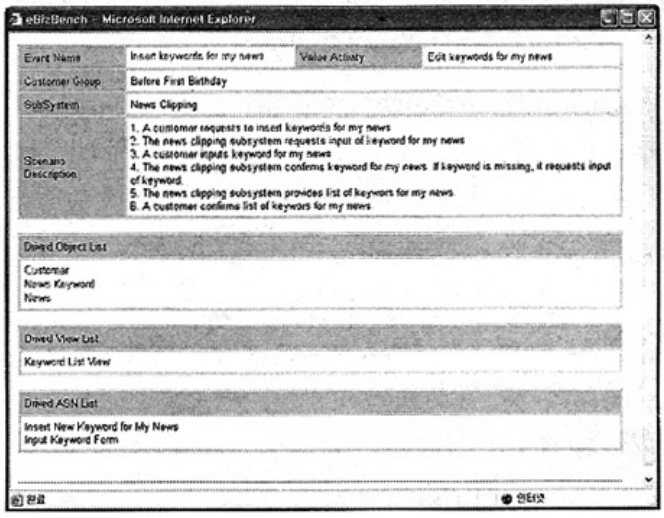

Figure 13-Screen of Meta-data Browser

Model Converter

The model converter provides the functionality to convert design results from one model into another in the semantic way. It refers the meta-model and the meta-schema of source model and target model.

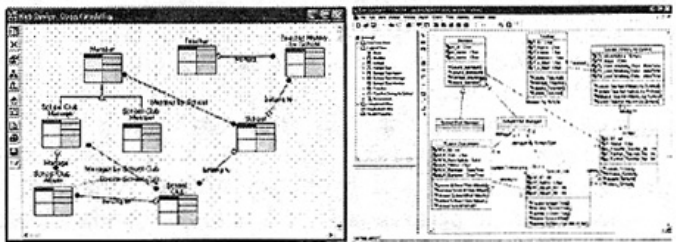

Figure 14-Conversion of Object Model to UML

Figure 14 shows the conversion of an object model (right screen) in the eBizBench into UML class diagram (left screen) in the Rational Rose. The object model, the class diagram, and the references between the object model and the class diagram should be predefined in the meta-model. The object model stored in the meta-schema can be converted into the class diagram according to references

stored in meta-model.

Comparisons of Support Systems for Developing Internet Business

This section compares the eBizBench with the other support systems for developing Internet business from the integrated perspective. Several researchers have proposed systems supporting for developing an Internet business system such as OOHDM-Web [61], Araneus [51, 52], AutoWeb [21], JWeb [25], and OO-H CASE Tool [28].

The OOHDM-Web [61] is an environment for the development of Internet business system, which uses the OOHDM [60]. It implements templates that are a mixture of HMTL and calls for functions in library giving access to the navigation objects stored in a relational database. The Araneus focused on the definition and prototype implementation of an environment for managing unstructured and structured Web contents. The AutoWeb [21] uses model-driven development to assist in the design and construction of Internet business. The JWeb is a system for design and prototyping of web applications to support HDM-driven design of hypermedia applications and definition of contents holding database. The OO-H CASE Tool [28] is developed to support OO-H method. The main focus of OO-H method is to capture all relevant information to create device-independent, front-end specifications.

For the comparison of these systems, several criteria are of interest: customer analysis, system requirement analysis, based data model, information modeling, navigation design, user interface design, implementation support, based methodology, meta-data, main purpose, maintenance support, portability, development tools, and components. The result of comparison is shown in Table 2.

The customer analysis is a starting point for system development [66]. The eBizBench provides the customer need analyzer module and the customer correspondence analyzer module in the customer analysis subsystem to systematic analysis of customers and their needs.

Ease of use and customers' satisfaction are important for a customer-oriented Internet business system [53]. The eBizBench adopts scenarios that can describe processes between customers and the Internet business system in the scenario analyzer on the web design subsystem. The use of scenarios is likely to enhance the usability of the Internet business system by reflecting customers' navigational requirements effectively.

The conceptual design prior to physical implementation is essential in information systems analysis and design. Internet business design is not an exception. The conceptual design in Internet business system consists of information modeling, navigation design, and user interface design. The OOHDM-Web uses conceptual model, navigation model, and Interface model according to OOHDM [60], which is the object-oriented methodology for developing hypermedia system. The Araneus uses ER design for information modeling and Araneus Data Model (ADM) for navigation design. The JWeb and the AutoWeb are based on hypermedia design model (HDM) [25], which is based on ER and consists of structure design, navigation design, and

Table 2- Comparisons of design and support system for developing Internet business system

Criteria \ Support Tools	OOHDM-Web	Araneus	AutoWeb	JWeb	OO-H CASE Tool	eBizBench
Reference	[60]	[51, 52]	[21]	[25]	[28]	This paper
Customer Analysis	N/A	N/A	N/A	N/A	N/A	- Customer Need Analysis - Customer Correspondence Analysis
System Requirements Analysis	N/A	N/A	N/A	N/A	N/A	Scenario
Based Data Model	OO	ER	ER	ER	OO	OO
Information Modeling	- Conceptual Model	- ER Design	- Structure Design	- Structure Design	- Class Diagram	- Object Modeling
Navigation Design	- Navigation View	- ADM Design	- Navigation Design	- Navigation Design	- NAD	- View Design - Navigation Design
User Interface Design	- Interface Model		- Presentation Design	- Presentation Design	- APD	- Page Schema Design
Implementation Support	- CGI	- Database - Web Page Template	- Database - Web Page	- Database - Web Page Template	- Page Template	- Database - Page Template - Simulation of Navigation
Based Methodology	OOHDM	ADM	HDM	HDM	OO-H Method	SOHDM
Meta-data	N/A	Design Repository	Meta-data	HDM-Schema	N/A	Repository
Main Purpose	Template-driven Design	Design and Development	Model-driven Design, Implementation, and Maintenances	Design, Reuse, Documentation, and Tools Communication	Device Independent Design and Implementation	Requirements Analysis, Design, Implementation and Reuse
Maintenance Support	N/A	Design Repository	Met-data	N/A	N/A	Repository
Portability	Not Portable	Portable	Portable	Portable	N/A	N/A
Development tool	CGI Scripts	Java	Java	Java	Not Specified	Visual Basic 6.0 and ASP
Components	- Authoring Environments - Browsing Environments - Maintenance Environments	- Ulixes - Editor - Minerva - ADM Object Manager - Penelope - Design Repository	- Diagram Editor - Relational Schema Generator - Prototype Generator - Style Sheet Editor - Page Generator - Data Entry Generator - Page Grabber - Administrator - Meta-Data	- Schema Editor - Mapper - Instance Editor - Configurator - Generator - HDM-Schema	- Editor - Model Compiler	- Project Manager - Customer Need Analyzer - Customer Correspondence Analyzer - Value Derivation Designer - Value Activity Modeler - Scenario Analyzer - Object Modeler - View and Navigation Designer - Page Specification Designer - DB Generator - Page Simulation Generator - Meta-Schema Manager - Meta-Data Manager - Model Convertor

presentation design. The OO-H CASE tool provides class diagram, navigational access diagram (NAD), and abstract presentation diagram (APD). The eBizBench uses object model, view design, navigation design, and page schema design for conceptual modeling based on SOHDM [45].

To support for implementing Internet business system, other systems generate CGI scripts, page temples, and database tables. The eBizBench generate not only database tables and page templates, but also simulation pages. Using simulation, usability of Internet business system to be constructed can be evaluated.

Because Internet business system continues to evolve and improve according to customers' needs. Internet business system development is not a one-time event, but a process with a long life cycle [27]. For these purpose, the Araneus and AutoWeb use design repository and meta-data, respectively. In the eBizBench, repository and repository subsystem are implemented not only for the reuse of design results to provide the basis of maintenance support, but also for conversion of design aspects to other design support systems.

The Araneus, AutoWeb, and JWeb are developed by the use of Sun Microsystems' Java for portability on different platforms. However, the eBizBench currently is not portable, because development tools for the eBizBench cannot provide portability. This weakness of eBizBench will be improved in next version developed by Java.

Conclusion

In Internet business environment, the speed of change is inconceivable. Because customers can compare products and services with a variety of rich information, they can easily move to new products or services [7]. Companies should analyze customers' experiences, and then respond to their needs [64, 65]. In these different bases of competition, Internet business system needs to be developed from customer-oriented perspectives.

This paper proposes a design support system, called eBizBench for developing customer-oriented Internet business system. The eBizBench focuses on the reuse and conversion of design result. It has three features different from those of the existing Internet business system design and development support systems; (i) its main focus is on semi-automatic support for the customer-oriented development of the Internet business system. It provides the customer analyzer, the customer correspondence analyzer, and scenario analyzer to reflect customers' needs in a systematic fashion. (ii) A repository and repository subsystem are implemented for the reuse and conversion of design result. A repository and repository subsystem can be used for the maintenance of Internet business system.

By using the eBizBench, companies are more likely to capture the context in which customers interact with their Internet business system in a systematic fashion. Repository and repository subsystem are better able to effectively support reusability, and thus enhance the maintenance of the system.

We are in the process of developing new version of eBizBench using Java. The major weakness of eBizBench, lack of portability will be overcome in the next version. On the basis of the current research, the following future research can be considered. First, automation of usability evaluation has several advantages [54]. Adoption of automated evaluation will effectively enhance the usability of Internet business system. Second, evaluation of customers' usage on Internet business system is important for reflecting customers' needs [67]. Web Mining is the extraction of interesting and potentially useful patterns. By adopting Web mining technique, customers' web usage can be effectively analyzed and then Internet business system can be improved for reflecting customers' needs.

Acknowledgments

The authors appreciate C.Y. Lee and J.K. Park for sharing information about A company.

References

[1] Aaron, R., Decina, M., and Skillen, R. (2000). "Electronic commerce: Enablers and implications," IEEE Communications Magazine, September, pp. 47-52.

[2] Aissi, S., Malu, P., and Srinivasan, K. (2002). "E-business process modeling: The next big step," *Computer*, Vol. 35, No. 5, pp. 55-62.

[3] Artz, J. M. (1996). "A top-down methodology for building corporate web applications," Internet Research: Electronic Networking Applications and Policy, Vol. 6, No. 2/3, pp. 64-74.

[21] Fraternali, P. and Paolini, P (2000). "Model-driven development of web applications: The Autoweb system," *ACM Transaction on Information Systems*, Vol. 28, No. 4, pp. 323-382.

[25] Garzotto, F., Paolini, P., and Baresi, L. (2001). "Supporting Reusable Web Design with HDM-Edit," *Proceedings of the 34th Hawaii International Conference on System Sciences*.

[28] Gómez, J. and Cachero, C. (2001). "Conceptual modeling of device-independent Web applications," *IEEE Multimedia*, Vol. 8, No. 2, pp. 26-39.

[51] Mecca, G., Merialdo, P., Atzeni, P., and Crescenzi, V. (1999). "The (short) Araneus guide to Web-site development," *Proceedings of the second International Workshop on the Web and Databases (WebDB'99) in conjunction with SIGMOD*.

[52] Merialdo, P., Atzeni, P., Magnante, M., Mecca, G., and Pecorone, M. (2000). "HOMER: a model-based CASE tool for data-intensive Web sites," *Proceedings of the SIGMOD*.

[60] Schwabe, D. & Rossi, G. (1995). "The object oriented hypermedia design model," *Communications of the ACM*, Vol. 38, pp. 45-46.

* Contact the authors for full references
(freeman@kgsm.kaist.ac.kr)