Developing Intranet Hypermedia System By Using Scenario-Based Object-Oriented Methodology

이 회 석 [1]
이 충 석 [2]
유 천 수 [2]
김 영 환 [2]
김 종 호 [2]
조 선 형 [3]

Abstract

In this paper, we develop an Intranet hypermedia system by using scenario-based object-oriented methodology. The methodology consists of six phases such as domain analysis, object modeling, view design, navigational design, implementation design and construction. Users' requirements are obtained by employing responsibility-driven object technologies. Views are generated from the resulting object model and then used for the subsequent navigational and implementation design. Implementation design phase deals with database schema, page structure and flow, and user interface. A hypermedia information system is built for a real-life commercial bank.

I. Introduction

Hypermedia systems are emerging as a futuristic class of complex information management systems. These systems allow users to share information from various media such as text, graphics, audio, and motion pictures. Already, the world enjoys its potential for structuring and accessing information because of the recent surge of World Wide Web (WWW) technologies for the Internet or Intranet [Frank, 1996]. As the use of hypermedia applications for enterprise in information processing increases, the research of their development methodology is active. For example, methodologies such as HDM (Hypermedia Design Method) [Garzotto et al., 1993], RMM (Relationship Management Methodology) [Isakowitz et al., 1995], EORM (Enhanced Object Relationship Model) [Lange, 1993], and OOHDM (Object Oriented Hypermedia Design Model) [Schwabe and Rossi, 1995] have been developed and expected to be used widely.

However, most past methodologies have several weaknesses. First, conceptual database design phase is relatively simple. Its relationship with hypermedia applications is not identified. Second, navigational patterns have been designed by the database schema only. Additional process-related information may be helpful. Furthermore, in the previous object-oriented approaches, the emphasis has been on implementation details rather than logical design. The expressive strengths of object modeling techniques may be found in the conceptual design phase.

[1] 한국과학기술원 테크노경영대학원 부교수
[2] 한국과학기술원 테크노경영대학원
[3] (주) 왕컴퓨터코리아 대표이사

This paper attempts to address the above design issues and proposes a scenario-based methodology for obtaining users' requirements (e.g., navigational and database requirements). The scenario is then designed in a form of object-oriented model. The methodology consists of six phase such as domain analysis, object modeling, view design, navigational design, implementation design, and construction.

II. Literature Review

There has been no formal design methodology associated with hypermedia approaches until the development of HDM [Bieber and Isakowitz, 1995]. Over 20 development groups in six different countries have used it for hypermedia applications in different domains. It is based on an object-oriented model that encourages the use of different perspectives in presenting the same conceptual entity in different ways.

RMM is based on E-R (Entity-Relationship) abstractions, and built on top of HDM. It enhances HDM with additional access structures (conditional indexes and guided tours) and proposes a seven step methodology for building hypermedia applications. RMM steps include E-R design, slice design, navigational design, conversion protocol design, user-interface design, runtime behavior design and construction/testing.

Object-oriented ideas have been used in the hypermedia field for several years. EORM was the first object-oriented design methodology. EORM consists of three design frameworks: class, composition and GUI (Graphical User Interface). Class framework deals with class identification and class refinement. The composition framework deals with composition identification and composition refinement. The GUI framework includes two activities; (i) presentation with window identification and (ii) mapping of classes and compositions to the presentations.

In OOHDM, a hypermedia application is built within the framework of a four-step process supporting an incremental or prototype model. In a domain analysis step, a conceptual model is built by the use of object-oriented technologies. OOHDM describes the navigational structure in terms of navigational contexts, which are induced from the navigation classes such as nodes, links, indexes, and guided tours. OOHDM builds an abstract interface model by defining perceptible objects in terms of interface classes. Implementation step maps interface objects to implementation objects.

Recently, a view based hypermedia design methodology (VHDM) has been proposed by Lee et al. [1995]. Views are users' perception of the hypermedia requirements. VHDM attempts to explore functionality of view in hypermedia applications.

III. A Scenario-Based Object-Oriented Design Methodology

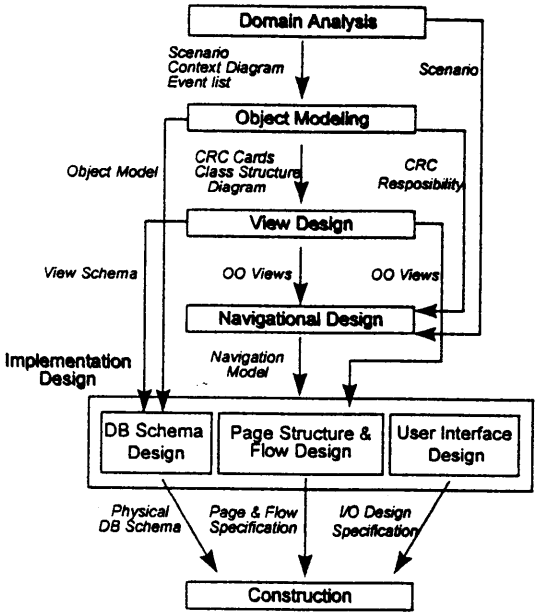
3.1 Architecture

This section describes the architecture of the scenario-based object-oriented hypermedia design methodology (SOHDM). SOHDM consists of domain analysis, object modeling, view design, navigational design, implementation design, and construction. The architecture of SOHDM is depicted in <Figure 3-1>.

Domain analysis phase adopts DFD (Data Flow Diagram) to represent system boundary and separate the target application system from external environments. In addition, scenario is introduced to identify user's requirements and navigational requirements. Scenario is a sequence of operations like use case. Object modeling phase models applications through the use of Taylor's technique [Taylor, 1995]. Object

oriented data model is designed through scenario sets. CRC (Class Responsibilities Collaborations) card and RDD (Responsibilities Driven Design) [Hutt, 1994] mechanisms are also used. In navigation design phase, developers design the paths that enable hypermedia navigation. Each associative relationship in the class diagram is analyzed. Scenarios are modified and then used to identify navigational requirements.

Implementation design phase defines users' information window (e.g., a single page of HTML documents) and information flow from one page to another page. Object model is transformed into relational database schema for database schema. It is important to note that our methodology is independent of physical database schema. In addition, I/O interface is designed. Finally, construction phase implements a physically running system.

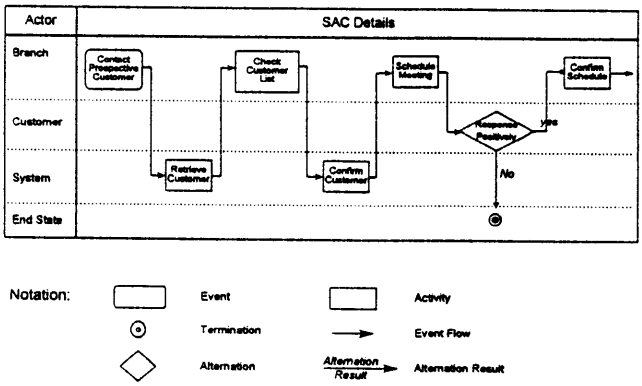


<Figure 3-1> Architecture of SOHDM

3.2 Design Phases

3.2.1 Domain Analysis

In domain analysis phase, SOHDM introduces scenario to identify hypermedia applications requirements. Scenario is similar to use case or script [Jacobson, 1995], but has very important differences. First, domain boundary is established. Domain boundary delimits the system to be developed. A context diagram of DFD is used.



<Figure 3-2> An Example of SAC

Next, scenario is created by the use of SAC (Scenario Activity Chart). SOHDM formalizes

requirements through the generation of operational scenarios that correspond to key business processes. EPC (Event Process Chain) [Young-Gul Kim, 1995] is slightly adjusted for SAC and then employed to formalize the scenario. EPC is originally investigated as a modeling tool for BPR (Business Process Reengineering). An example of SAC is given in <Figure 3-2>. Here, an actor generates operational processes. For further description of notations, readers are referred to [Lee, 1996].

During the scenario generation, it is important to add navigational requirements. A hypermedia system modeler is likely to find navigational path more effectively by using this SAC. As shown in <Figure 3-2>, information reference may be used for this purpose. This reference schema will be used for the subsequent navigational design phase.

3.2.2 Object Modeling

Scenario sets generated during domain analysis are used for object modeling. A CRC card is customized for our purpose. Also, RDD is adopted to gain active objects and model collaboration relationships among objects. Object modeling is performed as the following four sub-phases.

(i) Identify classes for each scenario: The first step is to look for the objects that are most essential to the primary scenarios. External objects and referenced information in scenario are candidates for objects. External objects play a role of actor, and therefore become active objects.

(ii) Produce CRC cards: Scenario should be transformed into objects. The transformation results in formatted index cards.

(iii) Define class structure (or relationships): There are three types of class relationship such as specialization, collaboration, and composition. For collaborations, developers place the names of the collaborators - the objects that are called upon for services - on the same line as the responsibilities they support in the CRC card. For superclasses, developers enter the name of the superclass. For composition, classes may have the attributes that refer to another class. These relationship can be described in the CRC card. In addition, CRC cards are transformed into class structure diagrams for implementation.

(iv) Review and update the model: Here, the design team walkthroughs the operation of the object model under various conditions. The model is reviewed for correctness and completeness. In detailing classes, developers begin by translating the responsibilities of their class into services. The service interface consists of all the methods that can be called upon by other classes. Once defined, these services are available to other objects.

3.2.3 View Design

In view design phase, information contents of domain classes in CRCs is reorganized as navigational units. This navigational unit is a view. In designing hypermedia applications, view support is highly desirable for a number of reasons. First, hypermedia applications should support a number of users, who have different requirements. Second, cognitive overhead can be reduced effectively. Because classes consist of a large number of heterogeneous attributes and methods, it is desirable to group attributes into views. Third, coherence is enhanced. Views can be used to represent semantic relationships among navigational nodes. Fourth, hypermedia applications can be easily extendible. Additional presentational or navigational requirements can be accommodated easily.

3.2.4 Navigational Design

In well-designed hypermedia applications, the way the users explore the hypermedia is an important design issue in order to avoid redundant information and prevent them from getting lost in the hyperspace [Schwabe, 1995]. Past studies base navigational design on data models in isolation. However, we believe that the use of scenario as well as data model improves the quality of the navigational design. We adopt

the following four user interface (UI) strategy as suggested by Shneiderman [1992]. Here, the root is the starting point in the page.

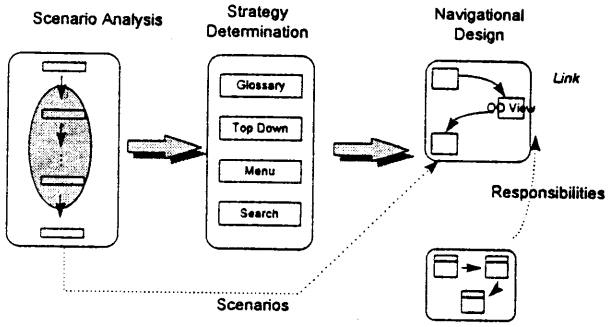
(i) Glossary: Makes the root that links to the related concept;

(ii) Menu: Organizes the root as a list or table of contents of the major concepts;

(iii) Top-down: Adopts a hierarchical menu;

(iv) Search: Generates a string search.

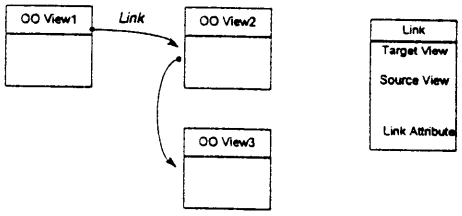
In summary, the detailed navigational design process is depicted in <Figure 3-3>.



<Figure 3-3> Navigational Design Process

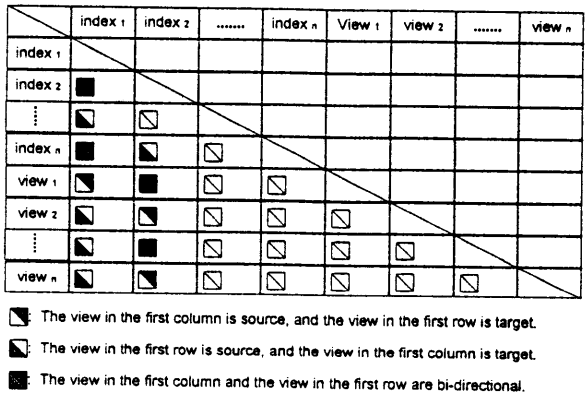
Scenario Types	UI Strategies
Not available from scenario	*Glossary*
Drill-down to explore information	*Top -Down*
Explore information with guide	*Menu*
Search information with key or through list	*Search*

<Table 3-2> Relationship between Scenario Types and UI Strategies



<Figure 3-4> Navigational Link

The first task in the navigational design is to determine UI strategies. These UI strategies may be referred to as access structure. It is noted that UI strategies can vary depending on a particular scenario. <Table 3-2> shows this relationship.



<Figure 3-5> Navigational Link Matrix

Next, navigational links are designed. In OOHDM, the navigational structure is defined by the use of

database schema that reflects the application domain. In SOHDM, however, OO views are navigational units. These OO views correspond to nodes in OOHDM. A page is built based on these OO views in the subsequent implementation design. A link consists of source view, target view, and link attributes. For example, if the source view is "Customer" and target view is "Account", then the corresponding attributes may be customer identification numbers.

For the sake of convenience, links may be summarized in a form of navigational link matrix. This matrix is illustrated in <Figure 3-5>.

3.2.5 Implementation Design

Implementation design phase generates schema, page structure, page flow, and user interface for construction. Hypermedia application can be developed under a variety of system environments. These include different DBMSs, and development tools such as CGI, HTML, JAVA[Gosling et. al 1995], or Shockwaves. However, our implementation design phase is transparent to these different system environments. Because relational database is still the most popular DBMS, system designers need rules to map object model (as denoted in class structure diagrams) into relational schema. These transformation rules can be found in [Blaha, 1988]. Furthermore in case of collaboration relationship, an additional view is generated.

Pages are then structured by organized OO views, anchors, and additional description details (text, image, sound, etc). Lastly, the page structure is enhanced to incorporate data location, interface components choice, and component properties. This enhancement result in the final user interface design.

IV. A Case

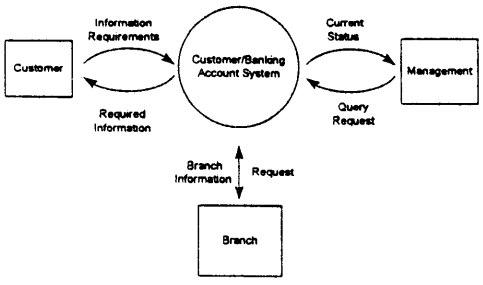
4.1 Case Description

Daegu Bank plans to implement WWW hypermedia via Intranet. Daegu Bank is one of the largest in South Korea. Its MIS system is implemented within the framework of a client/server architecture. However, some important applications are still subject to development. KAIST project team develops a prototype WWW hypermedia system for these applications.

HTML documents are generated and then connected to the file or database in Oracle[Oracle, 1995]. SOHDM is applied and the major system is implemented by the use of JAVA languages.

4.2 SOHDM Procedures

4.2.1 Domain Analysis



<Figure 4-1> Context Diagram

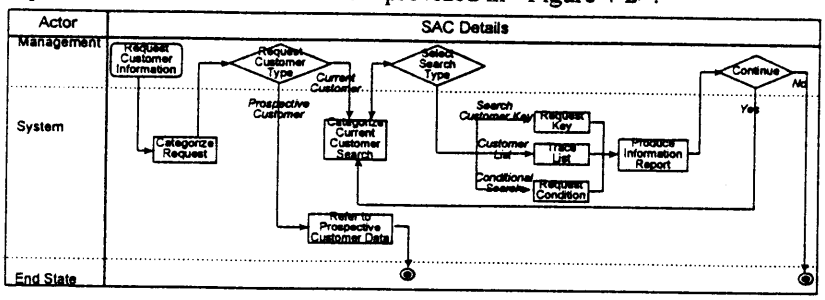
No	Event Name	Source
1	Request customer information	Branch
2	Request to create Account	Customer
3	Identify banking product types	Customer
4	Check customer account balance	Customer
5	Meet with prospective customer	Branch
6	Request branch input total	Branch/Management
7	Time to produce reports	Internal system

<Table 4-1> Event List

Application system boundary to be developed is established as shown in <Figure-12>. Customer, management, and branch are identified us external entities, and there are information flows between MIS system and these external entities. Next, for each external entity, events are trace. As shown in <Table 4-

1>, seven events are identified. Scenarios are generated from these events. Event 7 is temporal and others are generated by external entities.

Scenarios are main output during domain analysis phase. For the sake of presentation, a scenarios for "Request Customer Information" is provided in <Figure 4-2>.



<Figure 4-2> Scenarios - Request Customer Information

4.2.2 Object Modeling

Customer, account, and branch are identified as major objects and modeled in CRC card as shown in <Figure 4-3>. CRC card has been used effectively for finding and modeling objects. To store object's properties in the CRC card is simple and easy. Class structure diagram is produced from CRC cards as shown in <Figure 4-4>.

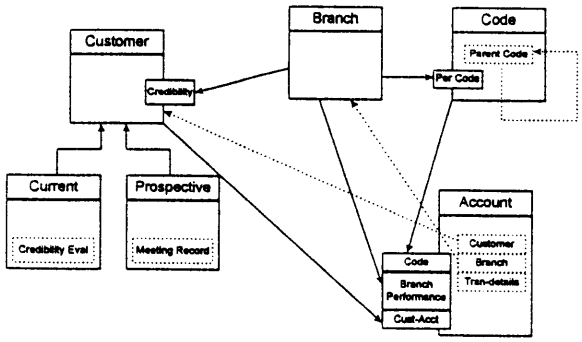
In producing class structure diagram, a class is associated with other classes by relationships in CRC card such as generalization in superclass, composition in components, and collaboration in collaborators.

Class : Customer	Superclass : Null
Subclass : Current Customer Prospective Customer	
Responsibilities	**Collaborators**
Check Balance Regist Account	Branch Account
Components	
Customer_Details, Customer_Assets	
Attributes List	
Customer_Id + Customer _Address + ...	

Class : Account	Superclass : Null
Subclass :	
Responsibilities	**Collaborators**
Manage account Process transaction	Branch Customer
Components	
Customer, Branch	
Attributes List	
Account_No + Customer_ID + Account_Type + ...	

Class : Branch	Superclass : Null
Subclass :	
Responsibilities	**Collaborators**
Calculate performance Manage customer Create account	Account Branch Customer
Components	
Parent-Branch	
Attributes List	
Branch_Code + Branch_Name + Branch_Address + ..	

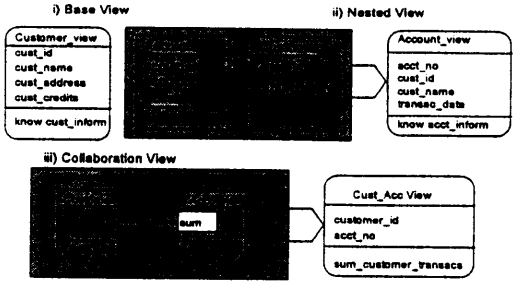
<Figure 4-3> Class Cards



<Figure 4-4> Class Structure Diagram

4.2.3 View Design

<Figure 4-5> is OO views derived from class structure diagram. OO views can be categorized into these types: base view, nested view, and collaboration view. A base view is made from a single class. A nested view is formed while developer traverse object to destination. A collaboration view is generated from collaboration relationship.



<Figure 4-5> Object Oriented View

4.2.4 Navigational Design

The first activity in the navigational design phase is to determine navigational strategies with scenarios. Scenario in <Figure 4-2> contains three navigational strategies: top-down, menu, and search strategies.

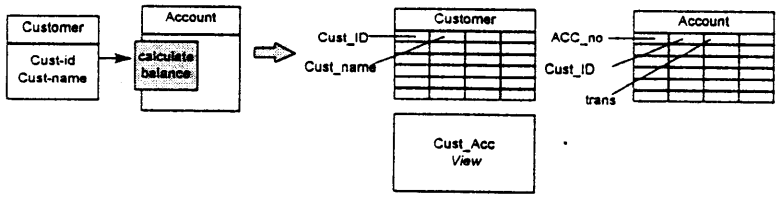
	Cust index	Trans Cust index	Search Cust with Key	List Cust	Search Cust with Condition	Cust_Acc View
Cust index						
Trans Cust index	■					
Search Cust with Key	◪	■				
List Cust	◪	■	◪			
Search Cust with Condition	◪	■	◪	◪		
Cust_Acc view	◪	■	■	■	■	

<Figure 4-6> Link matrix

From "Categorize request " to "Current Customer" and "Prospective Customer", Top-down strategy will be inferred. Menu strategy may be derived from "Select Customer Data Type", and search strategy is produced form "Use Customer Key" and "Conditional Trace". These access structures become parts of index pages. "Cust index" uses top-down, "Trans Cust index" uses menu, "Search Cust with key" and "Search Cust with" condition use search strategy.

Next step, formulates navigational links using index pages and views. Following link matrix in <Figure 4-6> is formed by index pages and views in scenario of <Figures 4-2>.

4.2.5 Implementation Design

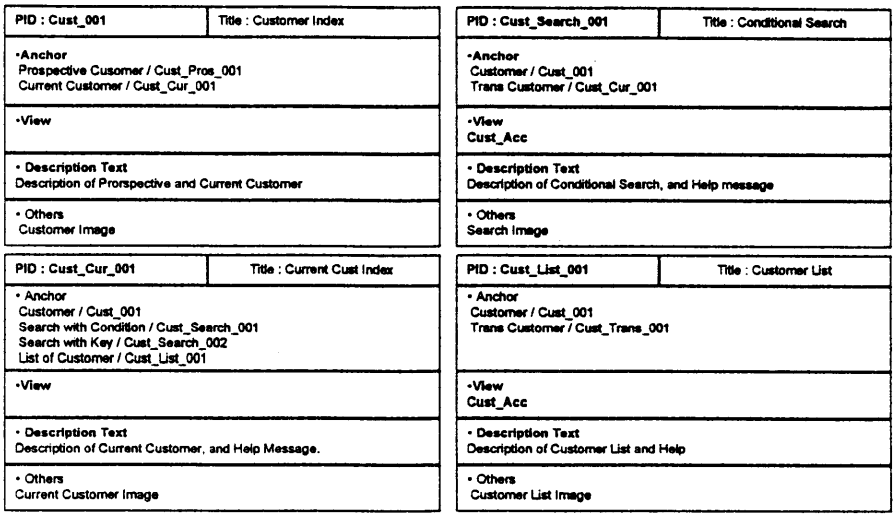


<Figure 4-7> Transformation Object Model into Relational Model

First subphase in implementation design, database schema design, transforms a class structure diagram into relational database schema. Database schema for implementation is consisted of base tables, user views, indices etc.. Following <Figure 4-7> shows an example of user view that calculates customer balance. In this subphase, all of these elements of database schema should be generated.

Next, implementation design phase performs generation of page structures and flows. In this subphase, page structures and flows are formed by using results of view design and navigation design phase.

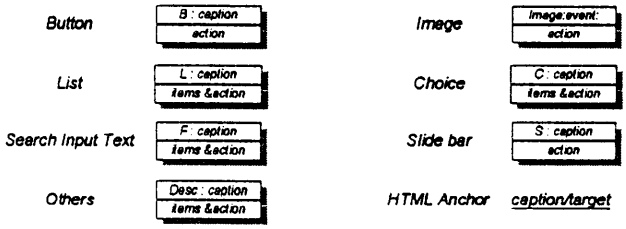
Following figures are samples of page structures and flows. These samples a group of information that users can retrieve from one page to others.



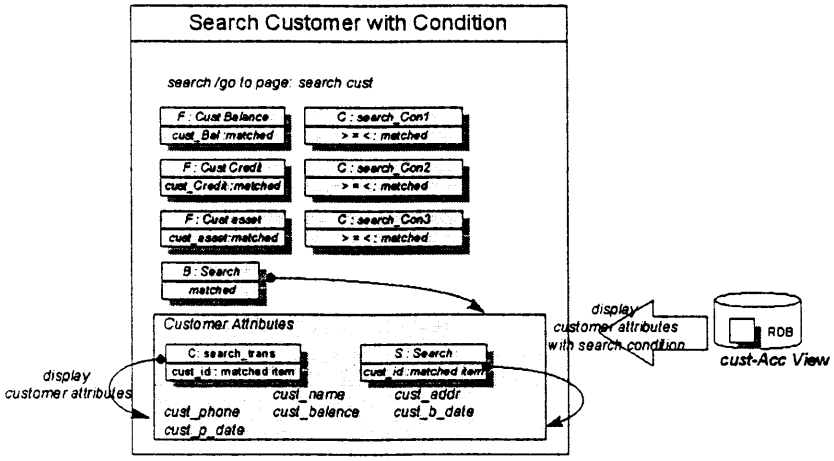
<Figure 4-8> Page Structure and Page Flow Schema

Third, user interface as a window of seeing information is required <Figure 4-9> is the notation used in user interface design. Users are able to gain required information by selecting and searching to be displayed contents on screen as shown in <Figure 4-11>.

Above these three products will be mapped to elements of target system to make a running system.



<Figure 4-9> Notation of User Interface Component



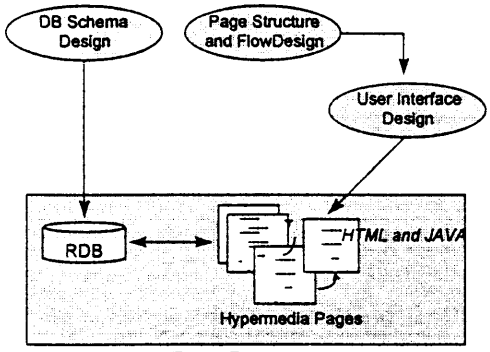
<Figure 4-10> User Interface Design Schema

4.2.6 Construction

In the construction phase, developers implement physically running hypermedia application system. All the products of implementation design phase should be mapped to physical elements as shown in <Figure 4-11>.

For example, database components such as tables, views, and indices corresponds to target relational DBMS, and they will be generated using facilities in such an environment. Also, page structures

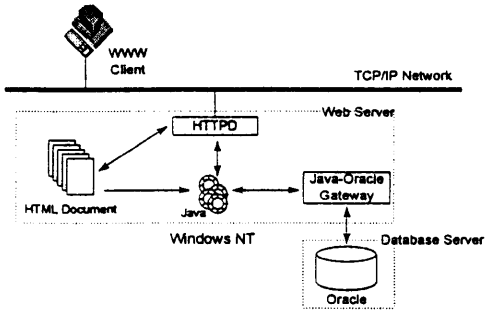
and flows, and I/O interface design map to HTML files and JAVA applets or scripts respectively. Since these three products of the implementation design phase are interrelated each other logically, if a target environment is established, these could be used faithfully for construction.



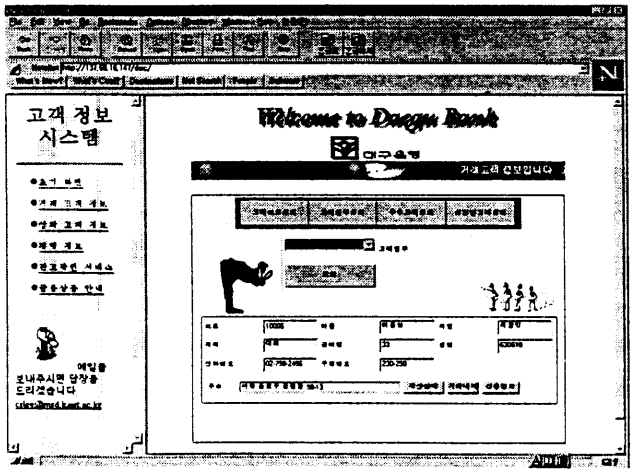
<Figure 4-11> Construction Process

4.3 Implementation Details

The implementation details are depicted in Figure 4-12. RDBMS plays a role of hyperbase and WWW is used as an information retrieval system within an Intranet framework. The Daegu Bank case is implemented by using Oracle. Java program is developed to insert database components into HTML documents. HTTP demon is a process which listens to requests from WWW clients and responds to them. From the main menu (Figure 4-13), "Customer Information System", we can navigate among the "Current Customer Information", "Prospective Customer Information", "Customer Management Information", and "Customer Support Service".



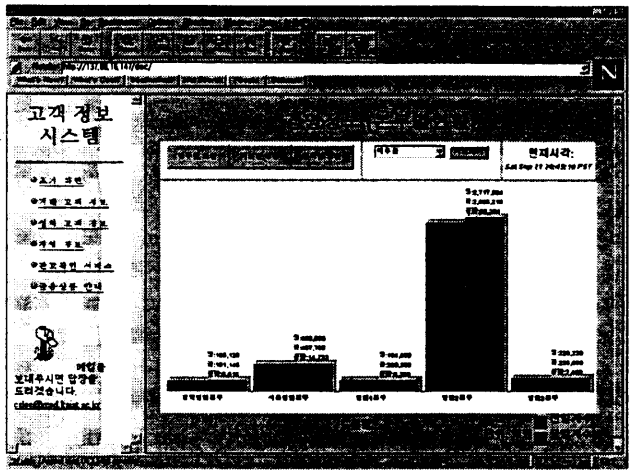
<Figure 4-12> Target Hypermedia System Architecture: WWW and RDBMS



<Figure 4-13> Current Customer Page

Clicking on the "Current Customer" anchor results in "Current Customer" page shown in Figure 4-13. This page has a scrolled text field which provides description of the current page, and four buttons such as "Key Search", "List", "Special Customer", and "Conditional Search". These buttons act as anchors leading users to another stages of the embedded applications.

Clicking on the "Customer MIS" anchor in main menu results in the "Customer MIS" page as shown in Figure 4-14. This page shows the amount of the accounts by division, the name of account, and both of them. This chart changes dynamically depending on the amount of money saved up to the particular moment of time. Currently, the total number of I/O pages are about 30.



<Figure 4-14> Customer MIS Page

V. Concluding Remarks

This paper proposed a scenario-based object-oriented methodology for developing an Intranet hypermedia applications. The methodology is the first to use scenario for capturing hypermedia requirements. The requirements are modeled in the form of object-oriented views and then transformed into page structures with navigational links. Adoption of scenario is more likely to improve the quality of navigational design. A real-life hypermedia system is developed to demonstrate the usefulness of our methodology.

The current research may be enhanced. A procedure to define scenarios can be formalized to obtain further users' requirements. In addition, a study of the use of methods as a realization of navigational relationships is of interest.

Acknowledgment

This research was partially supported by Wang Computer Korea Ltd. Research Grant GI90020.

References

[1] P. Balasubramanian, T. Isakowitz, and E. A. Stohr, *Designing hypermedia applications*, Proceedings of the 28th Hawaii International Conference on System Sciences, III, 1994, pp. 354-365.

[2] M. Bieber and C. Kacmar, *Designing hypertext support for computational applications*, Commun. ACM 38, 8 August 1995, pp. 99-107.

[3] M. R. Blaha, W. J. Premerlani, and J. E. Rumbaugh, *Relational Database Sesign using an Object-Oriented Methodology*, Commun. ACM, vol. 31, no. 4, April 1988.

[4] M. Frank, *Shifting gears*, Internet Systems, vol. 1, pp. 6-47, 1996.

[5] F. Garzotto, L. Mainetti, and P. Paolini, *Hypermedia design, analysis, and evaluation Issues*, Commun. ACM 38, 8 August 1995, pp. 74-86.
[6] F. Garzotto, P. Paolini, and D. Schwabe, *HDM: A model-based approach to hypertext application design*, ACM Trans. Off. Info. Syst. 11, 1 Jan. 1993, pp. 1-26.
[7] F. Garzotto, P. Paolini, and L. Mainetti, *Navigational patterns in hypermedia databases*, Proceedings of the 26th Hawaii International Conference on System Sciences, 1993, pp. 370-379.
[8] J. Gosling, and H. McGilton, *The Java Language Environment*, White Paper, Sun Microsystems Computer Company, October, 1995
[9] F. G. Halasz, and M. Schwartz, *The Dexter hypertext reference model*, Commun. ACM 37, 2 Feb. 1994, pp. 30-39.
[10] A. T. F. Hutt, *Object Analysis and Design - Description of Methods*, A Wiley-QED Publication, John Wiley & Sons, Inc. 1994.
[11] T. Isakowitz, E. A. Stohr and P. Balasubramanian, *Designing hypermedia applications*, Proceedings of the 27th Hawaii International Conference on System Sciences, 1994, pp. 354-365.
[12] T. Isakowitz, E. A. Stohr and P. Balasubramanian, *RMM: A Methodology for structured hypermedia design*, Commun. ACM 38, 8 August 1995, pp. 34-44.
[13] I. Jacobson, *Object-Oriented Software Engineering - A Use Case Driven Approach*, Addison-Wesley, 1995.
[14] Young-Gul Kim, *Process Modeling for BPR: Event-Process Chain Approach*, Proceedings of the Sixteenth International Conference on Information Systems, December 10-13, 1995, Amsterdam, The Netherlands.
[15] W. Kim, *Modern Database Systems - The Object Model, Interoperability, and Beyond*, Addison Wesley, 1995.
[16] D. B. Lange, *An Object-Oriented Design Approach for Developing Hypermedia Information Systems*, Journal of Organizational Computing, Forthcoming.
[17] D. B. Lange, *An Object-Oriented Design Method for Hypermedia Information Systems*, Proceedings of the 27th Hawaii International Conference on System Sciences, Maui, Hawaii, Vol. 3, January 4-7, 1994, pp. 336-375.
[18] D. B. Lange, *Enhanced Relationships in Object-Oriented Database Modeling*, Proceedings of InfoScience '93, Seoul, Korea, October 21-23, 1993, pp. 296-304.
[19] D. B. Lange, *Object-oriented hypermodeling of hypermedia supported information systems*, Proceedings of the 26th Hawaii International Conference on System Sciences, 1993, pp. 380-389.
[20] Heeseok Lee, Jongho Kim, Young Gul Kim, and Seon H. Cho, *A View Based Methodology for Designing Hypermedia Applications*, Proceedings of the 1996 Pacific Workshop on Distributed Multimedia Systems, Hong Kong, 27-28 Jun 1996. pp.101-110
[21] Choongseok Lee, *Implementing an Intranet Prototype by Using Scenario-Based Object Technology*, Master Thesis, KAIST, 1996.
[22] *Oracle 7 Server SQL Reference*, Release 7.2, Oracle Corporation, April 1995.
[23] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object Oriented Modeling and Design*, Prentice Hall, New York, 1991.
[24] D. Schwabe and G. Rossi, *Building hypermedia applications as navigational views of information models*, Proceedings of the 28th Hawaii International Conference on System Sciences, 1995, pp. 231-240.
[25] D. Schwabe and G. Rossi, *The object-oriented hypermedia design model*, Commun. ACM 38, 8 August 1995, pp. 45-46.
[26] D. Schwabe, G. Rossi, and S. D. J. Barbosa, *Abstraction, Composition, and Layout Definition Mechanisms in OOHDM*, Electronic Proceedings of the ACM Workshop on Effective Abstractions in Multimedia, Sanfrancisco, California, 4 November 1995.
[27] B. Shneiderman, *Designing the User Interface*, 2nd edition, Addison-Wesley Publishing Company .Inc., 1992.
[28] D. A. Taylor, *Business Engineering with Object Technology*, John Wiley & Sons, Inc., 1995.
[29] M. Thuring, J. Hannemann, and J. M. Haake, *Hypermedia and Cognition : designing for comprehension*, Commun. ACM 38, 8 August 1995, pp. 57-66.