USING SCENARIO FOR BUILDING HYPERMEDIA SYSTEMS

Choongseok Lee and Heeseok Lee

Corporate Information System Laboratory
Graduate School of Management, Korea Advanced Institute of Science and Technology
207-43, Chongryangri-dong, Dongdaemun-ku, Seoul, Korea, 130-012

ABSTRACT

This paper proposes a (SHDM, scenario-based hypermedia design methodology). The methodology consists of five phases: domain analysis, object modeling, navigation design, implementation design, and construction. Users' requirements are analyzed with a responsibility-driven technology using scenarios. Object-oriented views are generated as the result of object modeling, and then used for the subsequent navigation and implementation design. The implementation design phase deals with database schema, page structure and flow, and user interface. The SHDMTool is built to effectively support design phases. A repository based on IRDS standard is developed to integrate different hypermedia design methodologies. The SHDM integrates enterprise databases with distributed hypermedia systems such as internet, intranet, electronic commerce, and virtual enterprise.

KEYWORDS

Hypermedia, Scenario, Repository, Internet, Methodology

1. Introduction

Hypermedia extends the hypertext paradigm into multimedia. Hypertext is a nonsequential way of looking at text-based information [15]. Hypertext systems present text based documents including static data such as images and tables. This model (to which these notes adhere) closely resembles paper systems, with the added functionality of cross referencing between sections, usually by way of a keyword acting as an anchor, linked to another section. A Hypertext system then, consists of a network of nodes, connected via links.

While a variety of oppositions and advocates on hypermedia application can be found, we believe that hypermedia can fulfill its role as a medium for current and future information system.

There are a number of research issues related to the design, development, and the applications of hypermedia systems. Among these issues, we will focus on the design methodology and repository system for the following reasons.

First, hypermedia applications involve many different components, such as navigation, user interface, storage of content, and existence preparations. as a consequence, data models such as Data Flow Diagram (DFD), Entity-Relationship (E-R) diagram, and Object-Oriented hierarchies can not represent the design intricacies of hypermedia.

Second, many hypermedia developers have little experience in effectively incorporating hypermedia into their designs and implementations as numerous web pages attest.

Third, hypermedia, database, and their interactive environments that are important issues for the collaborative system are not sufficiently considered. there is no existing full-featured hypermedia design methodologies that one can use to develop different kinds of applications with different requirements.

The navigational and the user interface design are especially important in the hypermedia applications since the navigational design reflects user's requirement. Current research is that only the designer determines the navigation path using relationships in the data model. This is a weak approach in reflecting user's requirements. In this research, we present the scenario based approach for identifying user requirements (e.g., information and navigational requirements), object-oriented modeling using CRC card, and implementation design principles used in this methodology. Our methodology consists of five phases and its related issues on each phase, such as activities and products.

A hypermedia design methodology can not efficiently cover all hypermedia applications. Each hypermedia application needs proper hypermedia design methodologies. For this reason, this research will propose hypermedia design repository system. These can integrate and convert results of design of currently most effective methodologies

The followings are the primary objectives of this paper.

1. Proposing a new methodology for designing hypermedia applications with focus on navigational and implementation design by using users' requirements in early design phase.
2. Developing repository system for integrating different hypermedia design methodologies to cover all hypermedia applications.
3. Implementing the prototype that integrates WWW hypermedia and relational database system for virtual enterprise with interactive components.

2. Literature Review

Lately, the conceptual design methodology prior to implementation has been proven as an essential phase in information system analysis and design. Hypermedia design is not an exception. Several researchers have provided their methodologies on designing hypermedia applications. They include HDM by Garzotto, et al., RMDM by Isakowitz et al., EORM by Lange, OOHDM by Schwabe and Rossi, VHDM by H.Lee et al., IDM by Lee, SOHDM by H. Lee et al., and WHDM by H.Lee et al. Some methodologies are E-R based approaches and others are based on an object-oriented paradigm. In this section, we will review five frequently referenced methodologies: HDM, RMM, EORM, OOHDM, VHDM, IDM, SOHDM and WHDM.

We compared features of SOHDM with those of six other major hypermedia design methodologies. This comparison is summarized in Table 1.

Methodology Criteria	EORM [9]	RMM [7]	OOHDM [14]	VHDM [10]	IDM [12]	WHDM [11]	SHDM
Requirement Analysis	None	None	None	None	None	Workflow Analysis	Domain Analysis
Key Modeling Technique	OO	ER	OO	ER	Hierarchical Network	Workflow	OO
Source of Navigation	Object Relationship	Entity Relationship	Object Relationship	Entity Relationship	Meta Information Structure	Document Relationship	Scenario and Object Relationship
Approach to Identify users' view	None	Slice	View	View	Information Structure	Contents Analysis Cards	View
Collaborative Perspective	Communicative	Communicative	Communicative	Communicative	Communicative	Communicative Collaborative	Communicative Collaborative
Component Perspective	Moderate	Weak	Moderate	Weak	Weak	Weak	Strong
Users Perspective	Single	Single	Adaptive	Single	Single	Single	Adaptive
Support System	ODMTool	RMCase	None	None	None	HyDoMis	SHDMTool
Repository	None	None	None	None	None	Meta Data	IRDS-based Repository
Methodology Conversion	None	None	None	None	None	None	SHDMTool Conversion

Table 1. Comparison of Hypermedia Design Methodologies

3. A Methodology

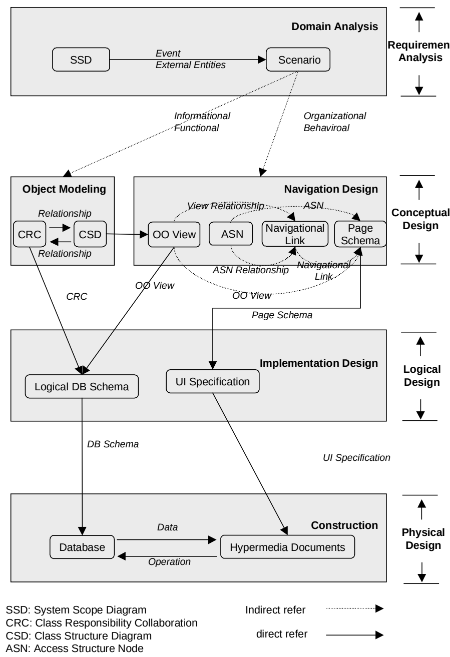
SHDM consists of domain analysis, object modeling, navigation design, implementation design, and construction. The architecture of SHDM is depicted in Picture 1.

Domain Analysis

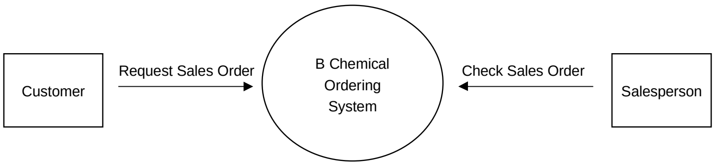
Domain analysis phase adopts System Scope Diagram (SSD) to represent system boundary and to separate the target application system from external environments. In addition, scenario is introduced to identify users' informational requirements and navigational requirements. A scenario is a sequence of operations like use case. In the object-modeling phase, applications are modeled using the Taylor's techniques [17] and object data model is designed through scenario sets. CRC (Class Responsibilities Collaborations) card and RDD (Responsibilities Driven Design) [6] mechanisms are also used. In the navigational 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.

we explain each phase of SHDM in further details, by using a real-life B chemical ordering system.

In the domain analysis phase, SHDM introduces scenario to identify hypermedia applications requirements. A scenario is similar to a use case or scripts [8], but they appear in different ways. In the first step of domain analysis, the domain boundary is established. Domain boundary delimits the system to be developed. A System Scope Diagram is used. For the design of B chemical ordering system, the system scope diagram is drawn as shown in Picture 2. This system has two external entities: customer and salesperson.



Picture 1. Methodology Architecture



Picture 2. System Scope Diagram

In the second step, scenario is created. SHDM formalizes requirements through the generation of operational scenarios that correspond to key business processes. Users and designers in users' perspective informally describe these scenarios between actors and system to be.

To describe scenario, a natural language approach is easy to describe and understand. Unfortunately, the natural language approach exhibits following several shortcomings [13].

First, natural language is lacking in formal syntax and semantics. In addition to the risk of ambiguity, the type and rigor of the analysis one may perform on informal requirements are clearly limited.

Second, there are currently no semantic approaches to analyzing dependencies among scenario and to detecting flaws. Incompleteness and inconsistency among scenarios must be resolved prior to advancing to the design phase.

SHDM uses scenarios to identify hypermedia applications requirements from the earliest opportunity by using natural language and then transformed to tabular format. In tabular form, scenarios are described in four process perspectives such as functional, behavioral, organizational, and informational which are suggested for process modeling in [3].

By using tabular, weakness of natural language can be supplemented and can be efficiently and effectively used in business domain. Scenarios are business processes according to actors. An actor is an operator of specific activities, i.e., a creator of events. External entities are the primary candidates for actors in scenario. An event is a starting point, a trigger of a scenario. An activity is an operation by which an actor completes a scenario. An activity has precondition and postcondition. These conditions are used to control activity flow. Precondition is the condition that is necessary to initiate activity. An activity is finished when postcondition is the satisfied. An activity flow is a sequence of activities. Termination is the end of a scenario. Information is required to activity. This information is described in information field in tabular. Activity is described in detail in description field.

During the scenario generation, it is important to add navigational requirements. In tabular form, designers describe previous scenarios in details in system perspective. Especially, interactions between actor and system are focused. Hypermedia application designers are likely to find navigational paths more effectively by using the scenarios. Interactions may be used for this purpose. Among the interactions determined by actor that create event, interactions related to the system are a primary concern for users' navigation since navigation is user dependent. 'Request sale order' is depicted in Table 2.

NO	Actor	Activity	Information	Precondition		Postcondition		Description
1	Customer	Connect to the ordering system	Customer					Start of the scenario
2	System	Request ID and password		User is connected				
3	Customer	Input ID and password	ID and password			ID and Password are inserted		
4	System	Validate ID and password	ID and password	ID and Password are inserted	2	ID and Password are correct	2	
5	System	Request Order Form	Customer	ID and Password are correct	2			
6	Customer	Input Order Form	Order Info					
7	System	Request Order Products	Order Info			Order form is validated	5	
8	Customer	Input products and quantity	Product					
9	System	Check Inventory on hand	Inventory	Products are Inserted	2	Inventory is on hand	5	
10	Customer	Confirm sale order	Sale order Information			Sale order is Confirmed	5	
11	System	Process sale order	Sale order Information	Sale order is Confirmed	7			End of the scenario

Table 2. Request sale order scenario

Object Modeling

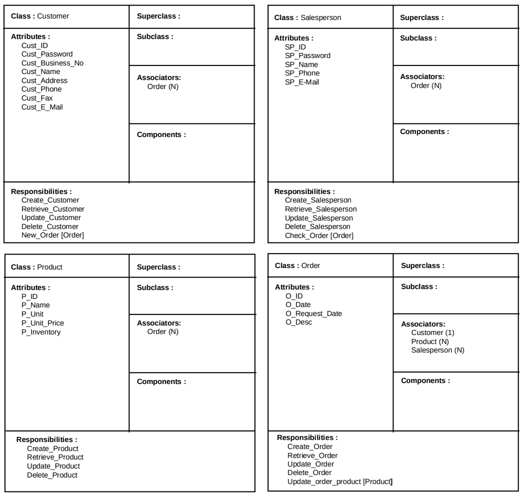
Scenarios in domain analysis phase are used for object modeling. Scenarios are transformed into objects in the form of CRC cards. The CRC cards are adopted because they have attractive informal appeal that helps make complex modeling tractable.

From scenarios in ordering system, four object classes (customer, salesperson, product, and order) are modeled. For example, customer class has eight attributes, one association relationship with order class, one collaboration relationship with order class, and five responsibilities.

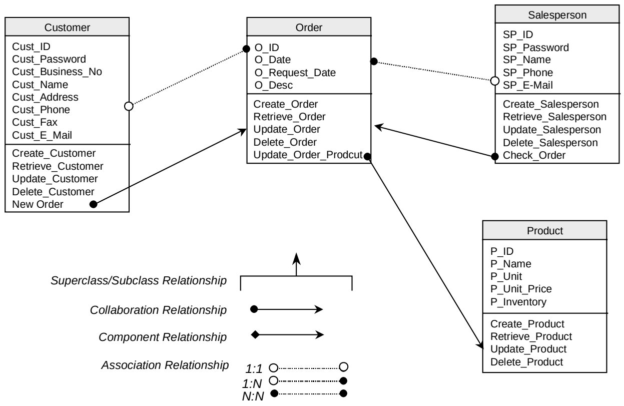
Four types of relationships are depicted in CRCs (superclass/subclass, collaborators, components, and associators). To present these relationships more effectively, a CSD (Picture 4) may be prepared.

Navigation Design

In the navigation design phase, first, information contents of domain classes in class structure diagram are reorganized as navigational units. Second, navigational paths and access structures are designed among navigational units. Third, Page Structures are designed on the base of navigational paths and access structures.
This navigational unit represents a view. The use of view in hypermedia application design has the following three advantages.
First, views can support a number of users who have different requirements.
Second, cognitive overhead can be effectively reduced, because heterogeneous attributes and responsibilities of objects are grouped into views.
Third, hypermedia applications are easily extendable, since additional requirements of presentation or navigation can be accommodated.

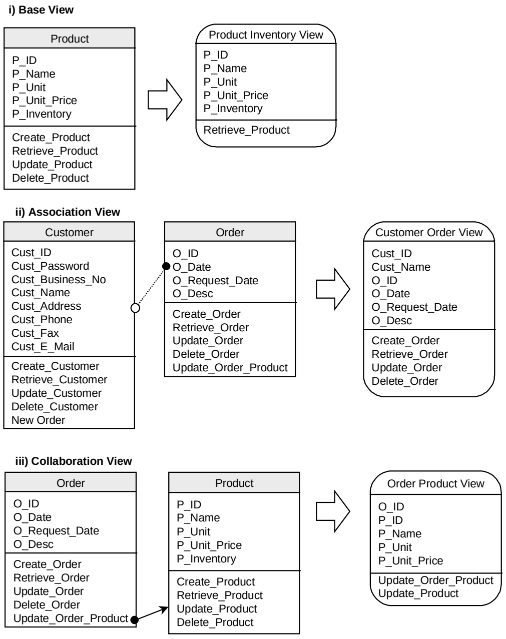


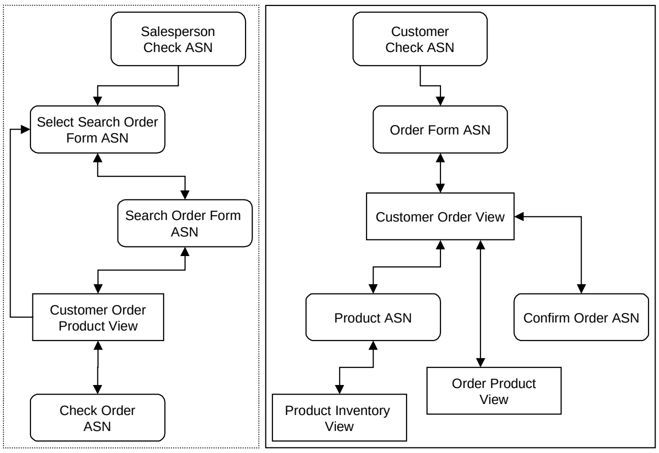
Picture 3. CRC Card



Picture 4. Class Structure Diagram

We can extract OO views on the basis of responsibilities and attributes in CRC cards as well as their relationship in CSD. Information perspectives in scenarios are primary candidates for designing OO views. OO views are categorized into three types: base view, association view, and collaboration view. A base view is generated from a single object class. An association view is extracted from an association relationship. Similarly, a collaboration view is generated from a collaboration relationship. These views may be referred to as object-oriented views (OO views). 3 OO views are described in Picture 5.





Picture 5. Generating Object Oriented Views

Picture 6. Navigational Link

In well-designed hypermedia applications, the way the users explore the hypermedia is an important design issue in order to avoid redundant information and to prevent them from getting lost in the hyperspace [14].

In navigational design phase, navigational paths and access structures are logically determined. Past studies only considered data models in navigational design. However, the use of scenario as well as data model will improve the quality of the navigational design.

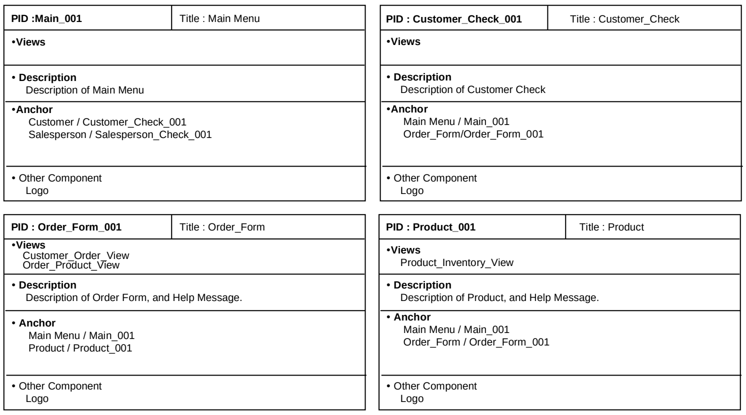
OO view and access structure node (ASN) are adopted for navigational units. ASN is used to implement the grouping, which allows users to access other parts of hypermedia documents in RMM. ASN provides users the access structures which users can use to navigate to different or detailed part of hypermedia application.

The second step in the navigation design phase is to determine ASNs by using scenarios; activity flows that start from the system to the actor are the primary concern for determining ASN. An activity from system to customer may become an ASN. The results of activity flow that is determined by postcondition in this activity may be the menu of the ASN.

Next, navigational links are built. The ANS differs from OO view in that ASN contains only access paths to OO views, but OO view contains the actual information that users want to obtain. These OO views and ASNs correspond to nodes. HTML pages are implemented on the basis of these OO views and ASNs in the subsequent implementation design. A link denotes the relationship between source and target node. This source and target node may be OO view or ASN. Navigational link is dipicted in Picture 6.

Finally, page structure and page flows are designed with the outputs from navigational and view design. A page is an information window that supplies for user information and navigational guide. The page structure design determines the information of a page particularly on view, anchor, text, image, and multimedia.

Page structures and flows are required for users to easily obtain necessary information, which are different for every user with discrete viewpoint. For this reason, many page structures may be created from a single view and have same ASN. On the other hand, a page structure may be composed of many views and many ASNs implying a many-to-many relationship in views and ASNs.



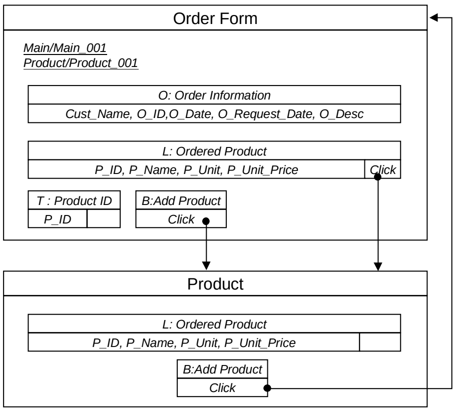
PID :Main_001 Title : Main Menu
- **Views**
- **Description**
  Description of Main Menu
- **Anchor**
  Customer / Customer_Check_001
  Salesperson / Salesperson_Check_001
- Other Component
  Logo

PID : Customer_Check_001 Title : Customer_Check
- **Views**
- **Description**
  Description of Customer Check
- **Anchor**
  Main Menu / Main_001
  Order_Form/Order_Form_001
- Other Component
  Logo

PID : Order_Form_001 Title : Order_Form
- **Views**
  Customer_Order_View
  Order_Product_View
- **Description**
  Description of Order Form, and Help Message.
- **Anchor**
  Main Menu / Main_001
  Product / Product_001
- Other Component
  Logo

PID : Product_001 Title : Product
- **Views**
  Product_Inventory_View
- **Description**
  Description of Product, and Help Message.
- **Anchor**
  Main Menu / Main_001
  Order_Form / Order_Form_001
- Other Component
  Logo

Picture 7. Page Schema



Picture 8. UI Specification

The ASN is implemented by index page, which supplies initial point to detailed information or other parts of hypermedia applications. For this reason, index page structure design is required first.

Pages are then structured by organized anchors, OO views, and additional description details (embedded components, text, images, sounds, etc.). Page schema is depicted in Picture 7.

Implementation Design

Implementation design phase generates logical database schema and user interface specification for construction. Hypermedia application can be developed under a variety of system environments. These include various DBMSs such as RDBMS or OODBMS, and development tools such as CGI, HTML, JAVA, [5], Shockwave, Flash, DHTML, XML, and SGML. However, implementation design phase in SHDM is transparent to these various system environments.

Database schema design step in design process is driving the model down to a low level of detail preparing it for implementation. The critical activity is translating each responsibility into methods that fulfill the responsibility. The collaborations that support each responsibility are expressed as messages to other objects within these methods. Finally, implementation decisions are made based on the requirements of the detailed design which include the choice of technology and the decision on how to optimally leverage target systems.

This methodology uses the CRC for object model and the relational database in implementing hypermedia application, therefore, transformation of object model into relational schema is required. The transformation may not be necessary if OODBMS is used. In many real-life cases, however, relational DBMSs are the most useful

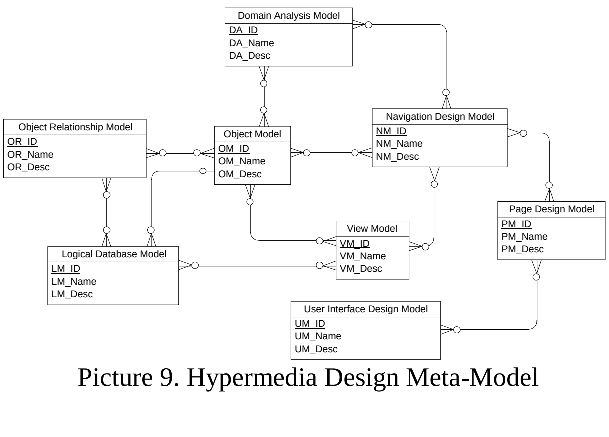
systems. Here, designers need rules that map object model into relational schema. The transformation rules can be found in [1].or [4]. We adapt Blaha's rule as follows:
First, each class is transformed into one table.
Second, a generalization relationship is transformed into one superclass table and multiple subclass tables.
Third, many-to-many relationships are transformed into distinct tables.
Finally, one-to-one and one-to-many relationships are transformed into distinct tables or merged with a participating class.
In addition to these rules, another transformation rule is considered for the collaboration relationships of our CRCs. Additional views or stored procedures may be required for these collaboration relationships.
For construction, the page structure should be enhanced to incorporate data location, interface components choice, and component properties.
Page schema in Picture 7 is enhanced to UI (User Interface) specifications in Picture 8 by the use of UI components.
These user interface components may be determined by access structure in the navigational design step. All access structures may be transformed into HTML anchors or set of HTML anchors. Unidirectional and bi-directional direct link may be transformed into buttons or images. Index, direct query and indexed query may be transformed into choice boxes or list boxes. Guided tour may be transformed into buttons which have "next" or "previous" caption or slide bars.

Construction

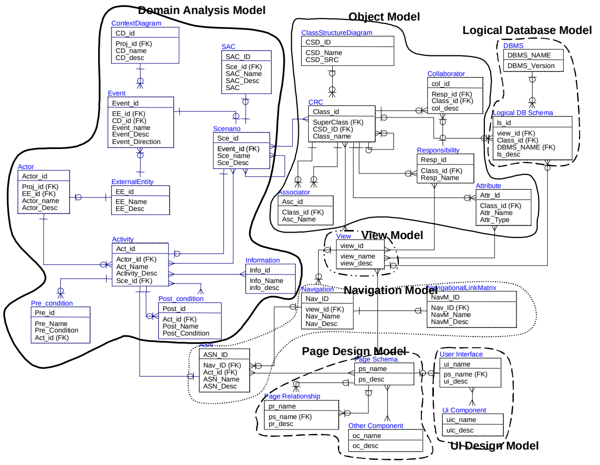
Result of the user interface design is used to construct hypermedia application in target environment- target DBMS and target implementation tools.
In the construction phase, developers implement a physically running hypermedia application system in target environments. All of the products during the implementation design phase should be mapped to physical elements.

4. Repository and SHDMTool

Several hypermedia design methodologies are suggested. Base models of these hypermedia design methodologies are categorized into E-R model and object model. RMM and VHDM are based on E-R model, while EORM, OOHDM, and SOHDM adopt an object model.
However, there is no existing full-featured hypermedia design methodologies that one can use to develop different kinds of applications with different requirements. Thus, if one wants to develop different applications, one might need to user different methodologies [2].
Hypermedia design conversion functionality would take as input the application requirements and transparently decide which methodology to use. It should provide the designer with the ability to move through the four basic design phases, such as object modeling, view design, navigational design, and user interface design, designing the application in a methodology independent way.
A repository is an integrated holding area. Despite the type of model being stored (logical data model, physical program structures, etc.), it can be related to other distinctly different model types that exist elsewhere in the repository. The contents should be definable, loadable, and retrievable regardless of originating environments [16].



Picture 9. Hypermedia Design Meta-Model



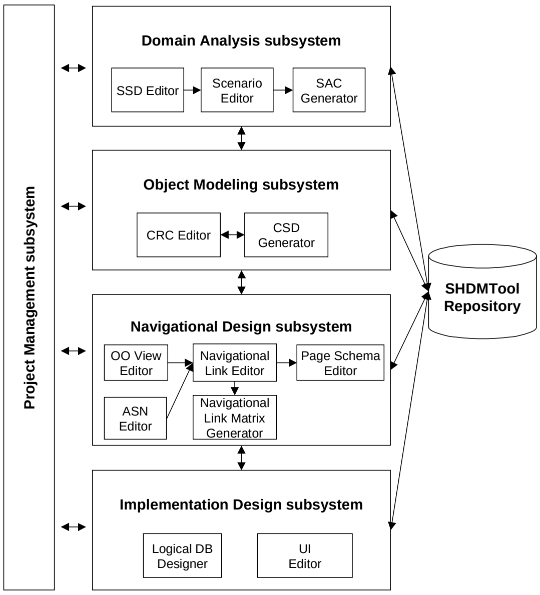
Picture 10. SHDM Meta-Schema

For hypermedia design conversion, hypermedia repository is needed. It provides the descriptive information of low level design output among different hypermedia design methodologies. We adopt IRDS 4 Level architecture for methodology conversion.

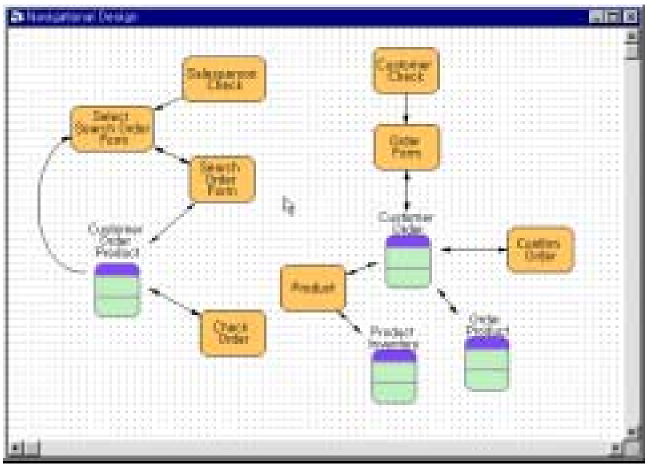
In IRDS 4 Level architecture, for hypermedia repository which is described in Picture 9, is IRD definition schema level. Picture 10 shows IRD definition level schema in SHDM. Picture 10 is the detail description of IRD definition schema level.

The domain analysis model means a hypermedia application domain requirement analysis models. Requirements in the domain are described by using many modeling methodologies, such as natural language, use case, scenario, petri-nets and so on. From these requirements, objects are elected and then described in object entity. Many object models such as hierarchical, network, E-R and OO can describe these objects. Object model has many object relationship models. Types of these relationships are summarized into object relationship model. Object views that can be derived from objects are described in view model. Object model and view model is converted to logical database model for implementation. Models such as navigational link matrix, hierarchical network, and so on are used to design hypermedia navigation. These models are described in navigation design model. page schema model and user interface model means the model for page schema and user interface.

SHDMTool has been developed to support hypermedia design and development phases as suggested by the SHDM. As seen in the previous chapter, SHDM can be manually, but only in small hypermedia application domain. As the range of domain is extended and the complexity is increased, however, it becomes almost impossible to analysis the domain and design hypermedia application by hand. A semi-automated system for supporting analysis and design is desired. This is why SHDMTool has been implemented. SHDMTool enhances the applicability of the SHDM. SHDMTool will continually improved to facilitate various aspects of SHDM. One of the directions is to accommodate the web-based distributed environment to facilitate the communication on the hypermedia requirements analysis, design, and implementation.



Picture 11. Architecture of SHDMTool



Picture 12. Navigational Design: Navigational Link Editor

SHDMTool has been implemented using the Microsoft Visual Basic 6.0 in the Windows 2000 environment and Microsoft SQL Server 7.0 in Windows NT 4.0 environment. IBM compatible Pentium PC has been used as a hardware platform.

SHDMTool has been designed as a tool to support the SHDM. It consists of six subsystems: 1) Project Management, 2) Domain Analysis, 3) Object Modeling, 4) Navigational Design, 5) Implementation Design, and 6) SHDMTool Repository. The overall architecture of SHDMTool is depicted in Picture 11.

In domain analysis phase, the user identifies system scope diagram and identifies scenarios, while SHDMTool is responsible for storing system scope diagram and scenarios and then generating scenario activity charts from scenarios. In object modeling phase, the task assigned to the user is designing object model in the form of CRC and that to SHDMTool is storing CRCs and generating class structure diagram. Designing OO view, ASN, and navigation link are tasks to the user in navigational design phase. In Implementation phase, the user designs logical database schema, phase schema, and UI specification. Picture 12 shows the screen of navigation link editor.

5. Conclusion

We proposed the SHDM methodology to design hypermedia application. The proposed methodology focus on electric commerce and virtual enterprise applications, which require capabilities such as collaborative, adaptive, and open hypermedia applications.

The SHDM focus on design for navigation and implementation by using requirements in early design phase. The SHDM consists of five phases such as domain analysis, object modeling, navigation design, implementation design, and construction. The use of scenario enhances the usability of hypermedia application by efficiently and effectively reflecting users' navigational requirements.

To efficiently support phases in SHDM, the SHDMTool is built. By using the SHDMTool, hypermedia design can be conveniently achieved. The methodology can be effectively and efficiently adjusted for complex hypermedia applications. A repository based on IRDS standard is developed to integrate different hypermedia design methodologies. The results of design of each methodology can be converted and then be used by other methodologies.

On the base of this research, we can consider the further research directions as follows.

First, requirement analysis is important, but many inconsistencies may be existing. Inconsistencies in scenarios can be detected by using petri-nets that provide mathematical foundation in process modeling. Scenario based on petri-nets can enhance to analyze users' requirement.

Second, system must integrate WWW and database for virtual enterprise and electronic commerce for the efficiency of SHDM and SHDMTool.

REFERENCES

1. Blaha, M. R., Premerlani, W. J. and Rumbaugh, J. E. (1988). Relational database design using an object-oriented methodology, Commun. ACM, 31(4)
2. Christodoulou, S. P., Stylinrus, G. D., and Papatheodorou, T. S. (1998). Evaluation of hypermedia application development and management systems, HyperText 98
3. Curtis, B., Kellner, M. I., and Jim Over (1992). Process modeling, CACM, 35(9), 75-90
4. Fong, J. (1995). Mapping extended entity relationship model to object modeling technique, SIGMOD Record, 24(3), 18-22
5. Gosling, J. and McGilton, H. (1995). The Java language environment, White Paper, Sun Microsystems Computer Company
6. Hutt, A. T. F. (1994). Object analysis and design - description of methods, A Wiley-QED Publication, John Wiley & Sons, Inc.
7. Isakowitz, T., Kamis, A., and Koufaris, M. (1997). Reconciling top-down and bottom-up design approaches in RMM, Proceedings of WITS-97, Atlanta, GA, USA
8. Jacobson, I. (1994). Object-oriented software engineering - a use case driven approach, Addison-Wesley.
9. Lange, D. B. (1994). An object-oriented design method for hypermedia information systems, Proceedings of the 27th Hawaii International Conference on System Sciences, Maui, Hawaii, 3, 336-375
10. Lee, H. Kim, J., Kim, Y. G., and Cho, S. H. (1996). A view based hypermedia design methodology, Proceeding of the 1996 Pacific Workshop on Distributed Multimedia Systems, Hong Kong , 101-110
11. Lee, H. and Suh, W. (1999). A workflow-based methodology for developing hypermedia information systems, The 5th International Conference of DSI (Decision Science Institute), Athens, Greece.
12. Lee, S. C. (1998). IDM: a methodology for Intranet design, Porceedings of the Nineteenth International Conference of Information Systems
13. Rolland, C. R. and Anchour, C. B. (1998). Guiding the construction of textual use case specifications, Data & Knowledge Engineering, 25, 125-160
14. Schwabe, D. and Rossi, G. (1998). An Object oriented approach to web-based application design, Theory and Practice of Object System, 4, 207-225.
15. Simon, A. (1995). Strategic database technology, Morgan Kumfmann Publishers, Inc.
16. Tannenbaum, A. (1994). Implementing a corporate repository, John Wiley & Sons, Inc.
17. Taylor, D. A. (1995). Business engineering with object technology, John Wiley & Sons, Inc.