A View-Based Hypermedia Design Methodology

Abstract

Hypermedia extends the hypertext paradigm into multimedia. Recently, many enterprises have attempted to incorporate hypermedia functionality into their diverse range of business applications such as decision support systems, document management, software management, and team collaboration. This paper presents a methodology for designing hypermedia applications based on users’ views. In designing hypermedia applications, view support is highly desirable. The methodology consists of five phases: requirement analysis, data modeling, view design, navigational design and mapping. Views are generated from the entity-relationship diagram, and used for subsequent navigational and mapping design. This methodology is especially effective for integrating enterprise databases with distributed hypermedia systems on the Internet.
I. Introduction

Traditional data management systems such as file system, relational databases and object oriented databases, have tried to articulate dynamic user requirements of multimedia, distribution, heterogeneity, documentation and bottom-up information infrastructure. These systems extend their functionality to cope with dynamically changing environments. Traditional information systems are, however, inflexible. They are storage-oriented systems with a sequential data management approach, and consequently cannot provide flexible information access as hypermedia systems do (Tompa 1989).

There have been a number of attempts to incorporate hypermedia functionality into a diverse range of applications, such as collaborative authoring of papers, decision support, document management, documentation, educational and corporate training, financial benchmarking, software engineering, and team collaboration (Bieber and Isakowitz 1995, Bieber and Kacmar 1995). Hypermedia extends the hypertext paradigm into multimedia. Figure 1-1 shows various issues involved with integrating hypermedia systems and traditional data management systems. Basically, these needs originate from the inability of traditional data management systems to cope with dynamically growing users’ requirements.

For designers, a simple and semantically rich model not only improves design and development processes, but also facilitates analyzing and evaluating hypermedia applications (Garzotto et. al 1995). “Yet, we lack guidelines and tools to design and develop hypermedia applications. Without such design guidelines and tools, the ever-growing network of interlinked applications is becoming increasingly spaghetti-like and hard to maintain” (Bieber and Isakowitz 1995, p26).

Conceptual design, prior to implementation, has proved to be an essential requirement in information system analysis and design. Hypermedia design is not an exception. Several researchers have supported this view. Therefore, two primary objectives of our paper are (i) to suggest a view-
based methodology for designing hypermedia applications called VHDM (View-Based Hypermedia Design Methodology), and (ii) to implement a real-life case, in which it integrates WWW (World Wide Web) hypermedia systems with a relational database by the use of VHDM.

![Diagram of Integration of Traditional Data Management Systems and Hypermedia Systems]

To our best knowledge, there had been little formal design associated with hypermedia approaches until the development of Garzotto, Paolini and Schwabe’s HDM (Hypermedia Design Method) (Garzotto et al. 1993, Kendall et al. 1992). Over 20 development groups in six different countries have used it for applications in different domains. It is based on an object-oriented model that builds hierarchical structures as an aggregation of simple ones, and at the same time encourages use of different perspectives in presenting the same conceptual entity in different ways.

A step-by-step methodology, RMM (Relationship Management Methodology) (Balasubramanian et al. 1994, Isakowitz et al. 1995) is based on an E-R approach. Familiarity with E-R abstractions among system analysts is well documented. RMM is built on top of HDM. It enhances HDM concepts with additional access structures (conditional indexes and guided tours) and proposes a seven step process 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 (Enhanced Object Relationship Model) (Lange 1993) was the first object-oriented design methodology; it consists of 3 frameworks: class framework, composition framework, and GUI framework. Two activities are related to the class framework, class identification and class refinement. The composition framework consists of a reusable library of link class definitions. The GUI framework includes two activities: (i) presentation with window identification and (ii) mapping of classes and compositions to the presentation.

In OOHDM (Object Oriented Hypermedia Design Model) (Schwabe and Rossi 1995), a hypermedia application is built within the framework of a four-step process supporting an incremental or prototype process model. In the domain analysis step, a conceptual model of the application domain is built by the use of object-oriented principles, augmented with some primitives associated with users and tasks. OOHDM describes the navigational structure of hypermedia applications in terms of navigational contexts, which are induced from the navigation classes such as nodes, links, indexes, and guided tours.

The original concept of VHDM was introduced by Lee et al. (1996a). A view is a subset of the hypermedia application domain associated with a particular user’s viewpoint. VHDM attempts to explore the functionality of view by optimizing various objectives of users, designers and developers. 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, it is desirable to group attributes into views. Third, views can be used to represent semantic relationships among navigational nodes. Fourth, additional presentational or navigational requirements can be accommodated easily (Thuring et al. 1995).
VHDM is based on the assumption that the practical target of hypermedia systems uses a relational database as information storage, for two reasons. The primary reason is that VHDM provides a mapping process between a hypermedia data model and actual hypermedia systems. Typically, relational database systems support views. Second, relational databases are used widely. This fact alone provides a motivation to explore VHDM. There is no doubt that hypermedia applications can help relational database systems to cope with a wave of changes with respect to IT (Information Technology) environment.

II. A View-Based Hypermedia Design Methodology

II.1 VHDM Architecture

We base our data model on RMM (Relationship Management Methodology). Our methodology is similar to RMM in that both are step-by-step methodologies, and start from E-R design. But RMM is a top-down approach; in contrast, VHDM (View-Based Hypermedia Design Methodology) is based on a bottom-up approach.

The phases are shown graphically in Figure 2-1. VHDM spans fundamental stages which include requirement analysis, conceptual modeling of a domain, and mapping of the results into the target hypermedia system. Hypermedia systems can be developed by a traditional SDLC (System Development Life Cycle) method, prototyping (Kendall and Kendall 1995), or both methods, depending on their characteristics.
II.1.1 Requirement Analysis
During the requirement analysis phase, the focus is on determining users’ needs, studying the application area in depth, assessing the strengths and weaknesses of the present work method, and reporting the results to management. In the traditional approach, the full requirements analysis is performed before the system design begins. In contrast, the requirements are analyzed as an ongoing procedure. Typically, users’ requirements can be analyzed by the use of natural languages, forms analysis, and file systems specifications (Batini et al. 1992). However, to accommodate the navigational requirement, additional sources may be needed.

II.1.2 E-R Design
In VHDM, the E-R diagram is used because it is quite familiar to many designers. The E-R diagram carries information about the structure of the domain, but, it does not specify all the navigational and access mechanisms. The VHDM methodology extends the initial E-R diagram into a hypermedia design diagram.
One of strengths of VHDM is associated with the fact that E-R design is popular and widely used. This fact can help designers use VHDM with the least amount of learning cost. For further details of E-R design, see, for example, (Batini et al. 1992). The design process represents a study of relevant entities and relationships of the application domain. These entities and relationships form the basis of the hypermedia application and many of them are expressed in the final application as nodes and links. In this paper, target hypermedia application is hypermedia interface to an existing database or a newly developed database.

II.1.3 View Design

View design is a process which reorganizes the information content of domain entities into navigational information units. View is a basic information unit. It may be used for navigation. This design phase results in view diagrams. Because hypermedia applications should support a number of users, who have different data requirements, the view design is important phase of the VHDM. Furthermore, because entities may consist of a large number of attributes of different natures, it is undesirable to present all of the attributes of an entity instance at once. Therefore, attributes are grouped into view. Accordingly, cognitive overhead can be reduced effectively.

A slice design in RMM is similar to this step. However, view design is different from slice design in that it allows overlap and cross reference over the entities. RMM presents attributes from referencing more than two slices. In RMM, collecting attributes by cross-referencing entities is also prohibited. Our methodology, however, is free from these two limitations. The benefits from releasing the prohibitions can be an enhancement of flexibility and coherence, and reduction of cognitive overhead.

II.1.4 Navigational Design

When views are made by collecting navigational units from entities, navigational design is followed. Navigational design establishes a navigational pattern of views. This design is supported in VHDM
by six access primitives, as shown in the Figure 2-2. The direct, indexed, guided tour and query navigation links are used to specify access between views. It is important to stress that if an anchor is a value, the associated link is a unidirectional or bi-directional link, and if an anchor is an attribute, the associated link is an indexed or guided tour or query navigation link.

[Figure 2-2] Primitives of Navigational Design

An index is a list of view instances providing direct access to each listed item. A guided tour links connects a collection of view instances in a linear sequence, with each instance connected to the next and previous one. In a circular collection, the last instance connects to the first. A guided tour navigation moves among members directly. If a view has a large number of instances, indexed links or guided tours will lose their functionality. In this situation, navigation by query is highly recommended. Query navigation has two kind of navigation types, indexed query and direct query. If a query result is the only one view instance, this link type will directly access the instance. If a query results in more than one instance, the link type will be changed into indexed link.

II.1.5 Mapping

The next step is to specify how the design results are realized in the target hypermedia application platform. This specification is done via a set of conversion rules between different data models. The goal of the mapping process is to translate the view and navigational design into a target hypermedia
system that is tailored to the specific hypermedia system. The mapping process is dependent on actual hypermedia system.

The mapping process consists of 5 substeps, which are graphically illustrated in Figure 2-3. The first step is mapping E-R schema into relational schema. Most designers have much experience in performing this step (Toerey 1994). In Step 2, view generation is performed according to the view design. Step 2 reflects information needs for specific user groups. During Step 3, a navigational index is generated. A navigational index has one-to-one correspondence with an indexed link. A navigational index can be pregenerated or dynamically configured. The choice is up to the developer. During Step 4, the users’ presentation requirements are satisfied. View and navigational index are flattened in the hypermedia document. Finally, a special system table called a link table is arranged for the purpose of effective application development and maintenance. Figure 2-4 shows an example of the link table.
II.2 Design Methodology Comparison

Garzotto, Mainetti and Paolini (1995) emphasized the importance of evaluation of the hypermedia applications, and proposed evaluation criteria such as richness, ease, consistency, self-evidence, predictability, readability and reuse. In Table 2-1, previously explained four methodologies and VHDM are compared.

RMM and VHDM are based on Entity-Relationship diagram while EORM (Enhanced Object Relationship Model) and OOHDM (Object Oriented Hypermedia Design Model) adopt an object-oriented approach. Accordingly, EORM and OOHDM become complex design methodologies, rich in semantics. In contrast, HDM (Hypermedia Design Method), RMM and VHDM are simple, but poor at semantics. The Entity-Relationship diagram provides a friendly interface to designers, and E-R based methodology is easy for the designers to use. The object-oriented approach is not helpful for designers in describing the user requirement or specifying a reality. Consistency can be enhanced if the methodology takes the sequence of contents along the navigational path into consideration. OOHDM and VHDM support consistency by providing a view mechanism. HDM, RMM and VHDM are more readable than the object-oriented approach, because of its ease of use. The
object-oriented approaches support reusability.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>HDM</th>
<th>RMM</th>
<th>EORM</th>
<th>OOHDM</th>
<th>VHDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Model</td>
<td>Hierarchical Network</td>
<td>Entity Relationship</td>
<td>Object Oriented</td>
<td>Object Oriented</td>
<td>Entity Relationship</td>
</tr>
<tr>
<td>Semantic Richness</td>
<td>Relatively Poor</td>
<td>Relatively Poor</td>
<td>Relatively Rich</td>
<td>Relatively Rich</td>
<td>Relatively Rich</td>
</tr>
<tr>
<td>Ease</td>
<td>Very Easy</td>
<td>Easy</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Easy</td>
</tr>
<tr>
<td>Consistency</td>
<td>Not Considered</td>
<td>Not Considered</td>
<td>Not Considered</td>
<td>Considered</td>
<td>Considered</td>
</tr>
<tr>
<td>Readability</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Reuse</td>
<td>Not Supported</td>
<td>Not Supported</td>
<td>Supported</td>
<td>Supported</td>
<td>Not Supported</td>
</tr>
</tbody>
</table>

Table 2-1 shows some dependencies among criteria. If the base model of a methodology is a hierarchical network or Entity-Relationship model, it is likely that the methodology is semantically poor and not supported in reuse, although easy to use and readable. In contrast, the object-oriented methodologies are semantically rich, and well supported in reuse. This fact indicates that basis of model determines the overall features of the methodology.

VHDM is based on the Entity-Relationship model. Accordingly, the advantages and disadvantages of E-R design are inherent in VHDM. Advantages are friendliness to designers, simplicity, readability and ease of use. However, VHDM misses dynamic and behavioral aspects of reality. In addition, VHDM is poor at semantics and cannot take advantage of reuse.

Originally, view is studied in database field for the purpose of supporting users who have various perceptions of database systems. Accordingly, employment of view in design process can enhance flexibility, coherence, extendibility, and security. These benefits of using views are transmitted to VHDM. Therefore VHDM is more extendible and flexible than any other.
methodology which does not use functionality of view as users define. In addition, VHDM is adaptable to both top-down and bottom-up approaches.

III. A Case

III.1 Case Description

A hypermedia application in KAIST (Korea Advanced Institute of Science and Technology) MIS (Management Information Systems) department needs information about departments, professors, courses, research papers, industrial projects, publications, special programs, etc. In 1996, KAIST ranks at the top among Korean universities in the evaluation made on the basis of faculty, research, management and library. The MIS department includes 10 faculty members and 150 graduate students. The hypermedia application is made up of several modules, which supports different types of users, each of them with different information needs. Possible types of users are visitors (to the department), students and faculty. Clearly, visitors have views different from those of students or faculty. To cope with the difficulties stemming from these differences, it is necessary to specify these different views.

III.2 Design Phases

The following are the design phases for the KAIST case.

Step 1. Requirement Analysis

For the data and navigational requirements, the form analysis, interview or file format analysis may be adopted. In our case, the MIS department pamphlet was analyzed first, and students and faculties were interviewed.

Step 2. E-R Design
Figure 3-1 shows an E-R diagram for the MIS department. The entities such as KAIST, faculty, students, publication, courses, laboratories, and projects are shown in rectangles. Attributes are attached to the corresponding entities. The relationships such as teaches, supervises, manages, publishes, is_opened_by, and performs are described.

Step 3. View Design

This step determines how the current entities can be reorganized to generate views. For example, Figure 3-2 shows the professor view. Collecting attributes from KAIST, faculty, course and lab entities results in the professor view. Figure 3-3 summarizes the professor view and other six views, which are employed for our prototype.
Step 4. Navigational Design

In this step, we design the navigational paths by defining anchors, links, indices, guided tours and query navigation. Anchors include a subset of attributes in a view. If a user activates anchors, he/she can navigate to another view through the link. Navigational methods vary, depending on the types of link. A direct link is used to navigate from one piece of information to another directly. This link is applied if the anchor has a particular value. Indexed navigation and guided tour are better if the destination is a view with a small number of view instances. If the destination is a view with a large number of instances, query navigation is more likely to be applied.
Figure 3-4 shows navigational paths among three views, professor, lab, and student. For example, if the student_name anchor in the lab view is activated, the student index is uploaded. The user then chooses one of the items in the student index and gets to the corresponding instance of the student view.

[Figure 3-4] A Partial Navigational Schema on the MIS Department Case

**Step 5. Mapping**

This step is highly dependent upon the target hypermedia system. In our KAIST case, the target hypermedia system is the integration of WWW with RDBMS. First, E-R schema is transformed into relational table(s). The result of this transformation is depicted in Table 3-1.

<table>
<thead>
<tr>
<th>Table</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>KAIST</td>
<td>Seoul_or_Taejon Address KAIST_image</td>
</tr>
<tr>
<td></td>
<td>History</td>
</tr>
<tr>
<td>Faculty</td>
<td>Faculty_name Professional_background</td>
</tr>
<tr>
<td></td>
<td>Research_Area E-mail_addr</td>
</tr>
<tr>
<td></td>
<td>Personal_Info Faculty_image</td>
</tr>
<tr>
<td></td>
<td>Phone_# Seoul_or_Taejon</td>
</tr>
<tr>
<td>Student</td>
<td>Student_ID Student_name</td>
</tr>
<tr>
<td></td>
<td>Student_Image Faculty_name</td>
</tr>
</tbody>
</table>
Next, we define view schema of KAIST MIS case. The detailed view schema is presented in Table 3-2.

<table>
<thead>
<tr>
<th>View</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>KAIST</td>
<td>Seoul_or_Taejon Address</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>History Image</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Professor</td>
<td>Faculty_name Professional_background</td>
</tr>
<tr>
<td></td>
<td>E-mail_addr</td>
</tr>
<tr>
<td></td>
<td>Faculty_image</td>
</tr>
<tr>
<td></td>
<td>Seoul_or_Taejon Course_name</td>
</tr>
<tr>
<td></td>
<td>Course_title</td>
</tr>
<tr>
<td></td>
<td>Publication</td>
</tr>
<tr>
<td>Student</td>
<td>Student_ID Student_name</td>
</tr>
<tr>
<td></td>
<td>Faculty_name Professor</td>
</tr>
<tr>
<td></td>
<td>Lab Professor</td>
</tr>
<tr>
<td>Publication</td>
<td>Title Journal</td>
</tr>
<tr>
<td></td>
<td>Year Faculty</td>
</tr>
<tr>
<td></td>
<td>Student</td>
</tr>
<tr>
<td>Lab</td>
<td>Lab_name Introduction</td>
</tr>
<tr>
<td></td>
<td>Faculty_name Lab</td>
</tr>
<tr>
<td></td>
<td>Project_name Lab_image</td>
</tr>
<tr>
<td></td>
<td>Project_org</td>
</tr>
<tr>
<td>Project</td>
<td>Project_name Student_name</td>
</tr>
<tr>
<td></td>
<td>Project_org Lab</td>
</tr>
<tr>
<td></td>
<td>Student</td>
</tr>
</tbody>
</table>
The navigational index is generated on the basis of the views. Table 3-3 shows the resulting navigational indices, relevant views, and attributes.

<table>
<thead>
<tr>
<th>View</th>
<th>Navigational Index</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professor</td>
<td>Professor Index</td>
<td>Faculty_name</td>
</tr>
<tr>
<td>Student</td>
<td>Student Index</td>
<td>Student_name</td>
</tr>
<tr>
<td>Course</td>
<td>Course Index</td>
<td>Course_name</td>
</tr>
</tbody>
</table>

[ Table 3-3] Definition of Navigational Index for KAIST MIS Case

Next, we turn to authoring hypermedia documents. Table 3-4 presents hypermedia documents as well as views and indices which are included in these documents.

<table>
<thead>
<tr>
<th>Hypermedia Document</th>
<th>Views and Indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professor</td>
<td>Professor View and Course Index</td>
</tr>
<tr>
<td>Professor Index</td>
<td>Professor Index</td>
</tr>
<tr>
<td>Publication</td>
<td>Publication Index and Publication View</td>
</tr>
<tr>
<td>Project Index</td>
<td>Project Index</td>
</tr>
<tr>
<td>Lab</td>
<td>Lab Index</td>
</tr>
<tr>
<td>Course</td>
<td>Course View and Course Index</td>
</tr>
</tbody>
</table>

[ Table 3-4] Definition of Hypermedia Document for KAIST MIS Case

**III.3 Implementation Details**

The KAIST MIS hypermedia system is actually implemented by the use of WWW and Sybase SQL
server (Mcgoveran and Date 1992). In Figure 3-5, RDBMS plays a role of hyperbase, and WWW is used as an information retrieval system in distributed environments. CGI (Common Gateway Interface) programs are developed in Perl language (Wall and Schwartz 1990). CGI programs insert database components into HTML (Hypertext Markup Language) documents. HTTPD (Hypertext Transfer Protocol Demon) is a process which listens to requests from clients and responds to them (December and Ginsburg 1995).

The system consists of 20 hypermedia documents. A hypermedia document is not always one-to-one correspondent with a view. A hypermedia document can contain multiple views and/or indices. For the sake of clear presentation, we show major 4 hypermedia pages as follows.

From the main menu page (Figure 3-6), one can get to the target information directly or start from the menu. That is, if 'Introduction to KAIST MIS' or 'MIS Faculty' anchors are activated, the corresponding index of the related information is uploaded. However, if an item in the drill-down menu is clicked, the target information page is selected directly. For other hypermedia pages, a number of paths are employed. Here, a page may be defined as a particular instance of hypermedia document.

Clicking 'MIS Faculty' anchor results in the faculty index as shown in Figure 3-7. The number of professors in KAIST MIS department is ten. Therefore, the query navigation may not be
necessary and thus an indexed link is adopted.

If an anchor of the indexed page is selected, the corresponding view instance is activated. For example, Figure 3-8 shows professor Heeseok Lee’s home page, an instance of the professor view. Anchors to the course views are included in the professor hypermedia page. If an anchor is clicked, an instance of the corresponding course view appears as shown in Figure 3-9.

IV. Conclusions

In this paper, we develop a view-based methodology for the design and implementation of
Employment of views in the hypermedia design methodology provides several benefits. Views support users’ different perceptions of hypermedia systems. Furthermore, they improve the maintenance of hypermedia information systems in terms of flexibility, coherence, and extendibility.

Sharpening our research requires further research. First, a systematic approach to capturing users’ navigational requirements is to be developed. This approach may need more than natural languages. Object modeling techniques may be good candidates (Lee et al. 1996b). Second, target systems other than WWW or RDBMS should be postulated. For example, OODB (Object-Oriented Database) can be used as a hyperbase system. Using OODB may improve the quality of hypermedia applications. Third, the applicability of data warehouse or data mining techniques in hypermedia development is to be further explored.

References


A View-Based Hypermedia Design Methodology

Heeseok Lee, Jongho Kim and Young Gul Kim
Korea Advanced Institute of Science and Technology, Korea

Seon H. Cho
Wang Computer-Korea

Hypermedia extends the hypertext paradigm into multimedia. Recently, many enterprises have attempted to incorporate hypermedia functionality into their diverse range of business applications such as decision support systems, document management, software management, and team collaboration. This paper presents a methodology for designing hypermedia applications based on users' views. In designing hypermedia applications, view support is highly desirable. The methodology consists of five phases: requirement analysis, data modeling, view design, navigational design and mapping. Views are generated from the entity-relationship diagram, and used for subsequent navigational and mapping design. This methodology is especially effective for integrating enterprise databases with distributed hypermedia systems on the Internet.

Traditional data management systems such as file system, relational databases and object-oriented databases, have tried to articulate dynamic user requirements of multimedia, distribution, heterogeneity, documentation and bottom-up information infrastructure. These systems extend their functionality to cope with dynamically changing environments. Traditional information systems are, however, inflexible. They are storage-oriented systems with a sequential data management approach, and consequently cannot provide flexible information access as hypermedia systems do (Tompa, 1989).

There have been a number of attempts to incorporate hypermedia functionality into a diverse range of applications, such as collaborative authoring of papers, decision support, document management, documentation, educational and corporate training, financial benchmarking, software engineering, and team collaboration (Bieber and Isakowitz, 1995, Bieber and Kacmar, 1995). Hypermedia extends the hypertext paradigm into multimedia. Figure 1 shows various issues involved with integrating hypermedia systems and traditional data management systems. Basically, these needs originate from the inability of traditional data management systems to cope with dynamically growing users' requirements.

For designers, a simple and semantically rich model not only improves design and development processes, but also facilitates analyzing and evaluating hypermedia applications (Garzotto et al., 1995). "Yet, we lack guidelines and tools to design and develop hypermedia applications. Without such design guidelines and tools, the ever-growing network of interlinked applications is becoming increasingly spaghetti-like and hard to maintain" (Bieber and Isakowitz, 1995, p.26).

Conceptual design, prior to implementation, has proved to be an essential requirement in information system analysis and design. Hypermedia design is not an exception. Several researchers have supported this view. Therefore, two primary objectives of our paper are (i) to suggest a view-based methodology for designing hypermedia applications called VHDM (View-Based Hypermedia Design Methodology), and (ii) to implement a real-life case, in which it integrates WWW(World Wide Web) hypermedia systems with a relational database by the use of VHDM.

To our best knowledge, there had been little formal design associated with hypermedia approaches until the development of Garzotto, Paolini and Schwabe's HDM (Hypermedia Design Method) (Garzotto et al., 1993, Kendall...
et al., 1992). Over 20 development groups in six different countries have used it for applications in different domains. It is based on an object-oriented model that builds hierarchical structures as an aggregation of simple ones, and at the same time encourages use of different perspectives in presenting the same conceptual entity in different ways.

A step-by-step methodology, RMM (Relationship Management Methodology) (Balasubramanian et al., 1994, Isakowitz et al., 1995) is based on an E-R approach. Familiarity with E-R abstractions among system analysts is well documented. RMM is built on top of HDM. It enhances HDM concepts with additional access structures (conditional indexes and guided tours) and proposes a seven step process 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 (Enhanced Object Relationship Model) (Lange, 1993) was the first object-oriented design methodology; it consists of three frameworks: class framework, composition framework, and GUI framework. Two activities are related to the class framework: class identification, and class refinement. The composition framework consists of a reusable library of link class definitions. The GUI framework includes two activities: (i) presentation with window identification and (ii) mapping of classes and compositions to the presentation.

In OOHD (Object Oriented Hypermedia Design Model) (Schwabe and Rossi, 1995), a hypermedia application is built within the framework of a four-step process supporting an incremental or prototype process model. In the domain analysis step, a conceptual model of the application domain is built by the use of object-oriented principles, augmented with some primitives associated with users and tasks. OOHD describes the navigational structure of hypermedia applications in terms of navigational contexts, which are induced from the navigation classes such as nodes, links, indexes, and guided tours.

Figure 1: Integration of Traditional Data Management Systems and Hypermedia Systems

The original concept of VHDM was introduced by Lee et al. (1996a). A view is a subset of the hypermedia application domain associated with a particular user's viewpoint. VHDM attempts to explore the functionality of view by optimizing various objectives of users, designers and developers. 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. Secondly, cognitive overhead can be reduced effectively. Because classes consist of a large number of heterogeneous attributes, it is desirable to group attributes into views. Thirdly, views can be used to represent semantic relationships among navigational nodes. Finally, additional presentational or navigational requirements can be accommodated easily (Thuring et al., 1995).

VHDM is based on the assumption that the practical target of hypermedia systems uses a relational database as information storage, for two reasons. The primary reason is that VHDM provides a mapping process between a hypermedia data model and actual hypermedia systems. Typically, relational database systems support views. Secondly, relational databases are used widely. This fact alone provides a motivation to explore VHDM. There is no doubt that hypermedia applications can help relational database systems cope with a wave of changes with respect to IT (Information Technology) environment.

A View-Based Hypermedia Design Methodology

VHDM Architecture

We base our data model on RMM (Relationship Management Methodology). Our methodology is similar to RMM in that both are step-by-step methodologies, and start from E-R design. But RMM is a top-down approach; in contrast, VHDM (View-Based Hypermedia Design Methodology) is based on a bottom-up approach.

The phases are shown in Figure 2. VHDM spans fundamental stages which include requirement analysis, conceptual modeling of a domain, and mapping of the results into the target hypermedia system. Hypermedia systems can be devel-
developed by a traditional SDLC (System Development Life Cycle) method, prototyping (Kendall and Kendall 1995), or both methods, depending on their characteristics.

**Requirement Analysis.** During the requirement analysis phase, the focus is on determining users’ needs, studying the application area in depth, assessing the strengths and weaknesses of the present work method, and reporting the results to management. In the traditional approach, the full requirements analysis is performed before the system design begins. In contrast, the requirements are analyzed as an on-going procedure. Typically, users’ requirements can be analyzed by the use of natural languages, forms analysis, and file systems specifications (Batini et al. 1992). However, to accommodate the navigational requirement, additional sources may be needed.

**E-R Design.** In VHDM, the E-R diagram is used because it is quite familiar to many designers. The E-R diagram carries information about the structure of the domain, but it does not specify all the navigational and access mechanisms. The VHDM methodology extends the initial E-R diagram into a hypermedia design diagram.

One of strengths of VHDM is associated with the fact that E-R design is popular and widely used. This fact can help designers use VHDM with the least amount of learning cost. For further details of E-R design, see Batini et al. (1992). The design process represents a study of relevant entities and relationships of the application domain. These entities and relationships form the basis of the hypermedia application and many of them are expressed in the final application as nodes and links. In this paper, target hypermedia application is hypermedia interface to an existing database or a newly developed database.

**View Design.** View design is a process which reorganizes the information content of domain entities into navigational information units. View is a basic information unit. It may be used for navigation. This design phase results in view diagrams. Because hypermedia applications should support a number of users, who have different data requirements, the view design is important phase of the VHDM. Furthermore, because entities may consist of a large number of attributes of different natures, it is undesirable to present all of the attributes of an entity instance at once. Therefore, attributes are grouped into view. Accordingly, cognitive overhead can be reduced effectively.

A slice design in RMM is similar to this step. However, view design is different from slice design in that it allows overlap and cross reference over the entities. RMM presents attributes from referencing more than two slices. In RMM, collecting attributes by cross-referencing entities is also prohibited. Our methodology, however, is free from these two limitations. The benefits from releasing the prohibitions can be an enhancement of flexibility and coherence, and reduction of cognitive overhead.

**Navigational Design.** When views are made by collecting navigational units from entities, navigational design is followed. Navigational design establishes a navigational pattern of views. This design is supported in VHDM by six access primitives, as shown in the Figure3. The direct, indexed, guided tour and query navigation links are used to specify access between views. It is important to stress that if an anchor is a value, the associated link is a unidirectional or bi-directional link, and if an anchor is an attribute, the associated link
is an indexed or guided tour or query navigation link.

An index is a list of view instances providing direct access to each listed item. A guided tour links a collection of view instances in a linear sequence, with each instance connected to the next and previous one. In a circular collection, the last instance connects to the first. A guided tour navigation moves among members directly. If a view has a large number of instances, indexed links or guided tours will lose their functionality. In this situation, navigation by query is highly recommended. Query navigation has two kind of navigation types, indexed query and direct query. If a query result is the only one view instance, this link type will directly access the instance. If a query results in more than one instance, the link type will be changed into indexed link.

**Mapping.** The next step is to specify how the design results are realized in the target hypermedia application platform. This specification is done via a set of conversion rules between different data models. The goal of the mapping process is to translate the view and navigational design into a target hypermedia system that is tailored to the specific hypermedia system. The mapping process is dependent on actual hypermedia system.

The mapping process consists of five substeps, which are graphically illustrated in Figure 4. The first step is mapping E-R schema into relational schema. Most designers have much experience in performing this step (Toorey 1994). In Step 2, view generation is performed according to the view design. Step 2 reflects information needs for specific user groups. During Step 3, a navigational index is generated. A navigational index has one-to-one correspondence with an indexed link. A navigational index can be pregenerated or dynamically configured. The choice is up to the developer. During Step 4, the users presentation requirements are satisfied. View and navigational index are flattened in the hypermedia document. Finally, a special system table called a link table is arranged for the purpose of effective application development and mainte-
nance. Figure 5 shows an example of the link table.

**Design Methodology Comparison**

Garzotto, Mainetti and Paolini (1995) emphasized the importance of evaluation of the hypermedia applications, and proposed evaluation criteria such as richness, ease, consistency, self-evidence, predictability, readability and reuse. In Table 1, previously explained four methodologies and VHDM are compared.

RMM and VHDM are based on Entity-Relationship diagram while EORM (Enhanced Object Relationship Model) and OOHDM (Object Oriented Hypermedia Design Model) adopt an object-oriented approach. Accordingly, EORM and OOHDM become complex design methodologies, rich in semantics. In contrast, HDM (Hypermedia Design Method), RMM and VHDM are simple, but poor at semantics. The Entity-Relationship diagram provides a friendly interface to designers, and E-R based methodology is easy for the designers to use. The object-oriented approach is not helpful for designers in describing the user requirement or specifying a reality. Consistency can be enhanced if the methodology takes the sequence of contents along the navigational path into consideration. OOHDM and VHDM support consistency by providing a view mechanism. HDM, RMM and VHDM are more readable than the object-oriented approach, because of its ease of use. The object-oriented approaches support reusability.

Table 1 shows some dependencies among criteria. If the base model of a methodology is a hierarchical network or Entity-Relationship model, it is likely that the methodology is semantically poor and not supported in reuse, although easy to use and readable. In contrast, the object-oriented methodologies are semantically rich, and well supported in reuse. This fact indicates that basis of model determines the overall features of the methodology.

VHDM is based on the Entity-Relationship model. Accordingly, the advantages and disadvantages of E-R design are inherent in VHDM. Advantages are friendliness to designers, simplicity, readability and ease of use. However, VHDM misses dynamic and behavioral aspects of reality. In addition, VHDM is poor at semantics and cannot take advantage of reuse.

Originally, view is studied in database field for the purpose of supporting users who have various perceptions of database systems. Accordingly, employment of view in design process can enhance flexibility, coherence, extendibility,
and security. These benefits of using views are transmitted to VHDM. Therefore, VHDM is more extendible and flexible than any other methodology which does not use functionality of view as users define. In addition, VHDM is adaptable to both top-down and bottom-up approaches.

A Case

Case Description

A hypermedia application in KAIST (Korea Advanced Institute of Science and Technology) MIS (Management Information Systems) department needs information about departments, professors, courses, research papers, industrial projects, publications, special programs, etc. In 1996, KAIST ranks at the top among Korean universities in the evaluation made on the basis of faculty, research, management and library. The MIS department includes ten faculty members and 150 graduate students. The hypermedia application is made up of several modules, which supports different types of users, each of them with different information needs. Possible types of users are visitors (to the department), students and faculty. Clearly, visitors have views different from those of students or faculty. To cope with the difficulties stemming from these differences, it is necessary to specify these different views.

Design Phases

The following are the design phases for the KAIST case.

Step 1. Requirement Analysis

For the data and navigational requirements, the form analysis, interview or file format analysis may be adopted. In our case, the MIS department pamphlet was analyzed first, and students and faculties were interviewed.

Step 2. E-R Design

Figure 6 shows an E-R diagram for the MIS department. The entities such as KAIST, faculty, students, publication, courses, laboratories, and projects are shown in rectangles. Attributes are attached to the corresponding entities. The relationships such as teaches, supervises, manages, publishes, is_opened_by, and performs are described.

Step 3. View Design

This step determines how the current entities can be reorganized to generate views. For example, Figure 7 shows the professor view. Collecting attributes from KAIST, faculty, course and lab entities results in the professor view. Figure 8 summarizes the professor view and other six views, which are employed for our prototype.

Step 4. Navigational Design

In this step, we design the navigational paths by defining anchors, links, indices, guided tours and query navigation. Anchors include a subset of attributes in a view. If a user activates anchors, he/she can navigate to another view through the link. Navigational methods vary, depending on the types of link. A direct link is used to navigate from one piece of information to another directly. This link is applied if the anchor has a particular value. Indexed navigation and guided tour are better if the destination is a view with a small number of view instances. If the destination is a view with a large number of instances, query navigation is more likely to be applied.

Figure 9 shows navigational paths among three views, professor, lab and student. For example, if the student_name anchor in the lab view is activated, the student index is uploaded. The user then chooses one of the items in the student index and gets to the corresponding instance of the student view.

Step 5. Mapping

This step is highly dependent upon the target hypermedia system. In our KAIST case, the target hypermedia system is
Figure 7: Definition of Professor View

Professor
- Faculty_Name
- Main_Career
- Research_Area
- E-Mail_Address
- Personal_Info
- Faculty_Image
- Phone_Num
- Seoul_or_Taejon
- Address
- Course_Name
- Lab_Name
- Publication

KAIST
- Seoul_or_Taejon
- Address
- History
- Image

Course
- Course_Number
- Course_Name
- Professor
- Course_Description

LAB
- Lab_Name
- Professor_Name
- Introduction
- Lab_Image
- Student_Name

Student
- Student_ID
- Student_Name
- Student_Image
- Publication
- Lab
- Professor
- Project

Publication
- Title
- Journal
- Year
- Faculty
- Student

Project
- Project_name
- Project_org
- Project_year
- Lab
- Professor
- Student

Figure 8: View Schema for MIS Department Case

Figure 9: A Partial Navigational Schema on the MIS Department Case
the integration of WWW with RDBMS. First, E-R schema is transformed into relational table(s). The result of this transformation is depicted in Table 2.

Next, we define view schema of KAIST MIS case. The detailed view schema is presented in Table 3.

The navigational index is generated on the basis of the views. Table 4 shows the resulting navigational indices, relevant views, and attributes.

Next, we turn to authoring hypermedia documents. Table 5 presents hypermedia documents as well as views and indices that are included in these documents.

**Table 2: E-R Schema for KAIST MIS Case**

<table>
<thead>
<tr>
<th>View</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>KAIST</td>
<td>Seoul_or_Taejon</td>
</tr>
<tr>
<td></td>
<td>History</td>
</tr>
<tr>
<td></td>
<td>Faculty_name</td>
</tr>
<tr>
<td></td>
<td>Research_Area</td>
</tr>
<tr>
<td></td>
<td>Personal_Info</td>
</tr>
<tr>
<td></td>
<td>Phone_#</td>
</tr>
<tr>
<td>Student</td>
<td>Student_ID</td>
</tr>
<tr>
<td></td>
<td>Student_Name</td>
</tr>
<tr>
<td>Publication</td>
<td>Title</td>
</tr>
<tr>
<td></td>
<td>Year</td>
</tr>
<tr>
<td>Faculty</td>
<td>Faculty_name</td>
</tr>
<tr>
<td></td>
<td>Lab_name</td>
</tr>
<tr>
<td></td>
<td>Lab_image</td>
</tr>
<tr>
<td>Project</td>
<td>Project_name</td>
</tr>
<tr>
<td></td>
<td>Project_year</td>
</tr>
<tr>
<td>Course</td>
<td>Course_name</td>
</tr>
<tr>
<td></td>
<td>Course_description</td>
</tr>
<tr>
<td>Is_opened_by</td>
<td>Course_number</td>
</tr>
<tr>
<td>Performs</td>
<td>Lab_name</td>
</tr>
</tbody>
</table>

**Table 3: View Schema of KAIST MIS Case**

<table>
<thead>
<tr>
<th>View</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>KAIST</td>
<td>Seoul_or_Taejon</td>
</tr>
<tr>
<td></td>
<td>History</td>
</tr>
<tr>
<td></td>
<td>Faculty_name</td>
</tr>
<tr>
<td></td>
<td>Research_Area</td>
</tr>
<tr>
<td></td>
<td>Personal_Info</td>
</tr>
<tr>
<td></td>
<td>Phone_#</td>
</tr>
<tr>
<td></td>
<td>Address</td>
</tr>
<tr>
<td></td>
<td>E-mail_addr</td>
</tr>
<tr>
<td>Student</td>
<td>Student_ID</td>
</tr>
<tr>
<td></td>
<td>Student_name</td>
</tr>
<tr>
<td></td>
<td>Lab</td>
</tr>
<tr>
<td></td>
<td>Project</td>
</tr>
<tr>
<td></td>
<td>Title</td>
</tr>
<tr>
<td></td>
<td>Year</td>
</tr>
<tr>
<td></td>
<td>Lab</td>
</tr>
<tr>
<td></td>
<td>Lab</td>
</tr>
<tr>
<td></td>
<td>Lab</td>
</tr>
<tr>
<td></td>
<td>Lab</td>
</tr>
<tr>
<td></td>
<td>Project</td>
</tr>
<tr>
<td></td>
<td>Project</td>
</tr>
<tr>
<td></td>
<td>Professor</td>
</tr>
<tr>
<td></td>
<td>Journal</td>
</tr>
<tr>
<td></td>
<td>Faculty</td>
</tr>
<tr>
<td></td>
<td>Professor</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
</tr>
<tr>
<td></td>
<td>Faculty_name</td>
</tr>
<tr>
<td></td>
<td>Project.org</td>
</tr>
</tbody>
</table>

**Implementation Details**

The KAIST MIS hypermedia system is actually implemented by the use of WWW and Sybase SQL server (McGeveran and Date 1992). In Figure 10, RDBMS plays a role of hyperbase, and WWW is used as an information retrieval system in distributed environments. CGI (Common Gateway Interface) programs are developed in Perl language (Wall and Schwartz 1990). CGI programs insert database components into HTML (Hypertext Markup Language) documents. HTTPD (Hypertext Transfer Protocol Demon) is a process which listens to requests from clients and responds to
them (December and Ginsburg 1995).

The system consists of 20 hypermedia documents. A hypermedia document is not always one-to-one correspondent with a view. A hypermedia document can contain multiple views and/or indices. For the sake of clear presentation, we show four major hypermedia pages as follows.

From the main menu page (Figure 11), one can get to the target information directly or start from the menu. That is, if 'Introduction to KAIST MIS' or 'MIS Faculty' anchors are activated, the corresponding index of the related information is uploaded. However, if an item in the drill-down menu is clicked, the target information page is selected directly. For other hypermedia pages, a number of paths are employed. Here, a page may be defined as a particular instance of

---

![Table 4: Definition of Navigational Index for KAIST MIS Case](image)

![Figure 10: Target Hypermedia System Architecture](image)

![Figure 11: Main Menu of KAIST MIS Case](image)

![Figure 12: Faculty Index](image)

![Figure 13: An Instance of Faculty View](image)

---

`Journal of Database Management`
Conclusions

In this paper, we develop a view-based methodology for the design and implementation of hypermedia applications, and illustrate its practical usefulness through a real-life case. Our methodology enhances the existing RMM (Relationship Management Methodology) and adds functionality of views. The target hypermedia system integrates WWW (World Wide Web) with RDBMS (Relational Database Management Systems).

Employment of views in the hypermedia design methodology provides several benefits. Views support users' different perceptions of hypermedia systems. Furthermore, they improve the maintenance of hypermedia information systems in terms of flexibility, coherence, and extendibility.

Sharpening our research requires further research. First, a systematic approach to capturing users' navigational requirements is to be developed. This approach may need more than natural languages. Object modeling techniques may be good candidates (Lee et al., 1996b). Second, target systems other than WWW or RDBMS should be postulated. For example, OODB (Object-Oriented Database) can be used as a hyperbase system. Using OODB may improve the quality of hypermedia applications. Third, the applicability of data warehouse or data mining techniques in hypermedia development is to be further explored.

References


Heeseok Lee is an associate professor of MIS at Korea Advanced Institute of Science and Technology. He received his bachelor's degree in Industrial Engineering from the Seoul National University, Korea, his master's degree in Industrial Engineering from the Korea Advanced Institute of Science and Technology, and his doctorate in Management Information Systems from the University of Arizona. He taught MIS for the university of Nebraska at Omaha during 1992-1994. Dr. Lee's research interests include enterprise databases, distributed objects, data warehouse and data mining, client/server, and hypermedia. He has had research papers published in Journal of Database Management, Telematics and Informatics, Information and Management, Omega, European Journal of Operational Research, Information Processing and Management, Journal of Systems and Software, Information Systems, Computers and Operations Research, Computers and Industrial Engineering, Systems Science, and Annals of OR.

Jongho Kim is a doctoral candidate in Management Information Systems at the Korea Advanced Institute of Science and Technology. He holds a bachelor's degree and a master's degree from Korea Advanced Institute of Science and Technology. His research interests include design methodology for hypermedia applications, management of hypermedia projects, and software process. He is now participating in the development of an enterprise resource management system incorporating hypermedia functionality.

Young Gul Kim is an associated professor of MIS at Korea Advanced Institute of Science and Technology. He received a Ph. D. in Management Information Systems from the University of Minnesota in 1990. Dr. Kim taught MIS for the University of Pittsburgh during 1990-1993. His main research interests include IS architecture, strategic management of IS resources, data modeling and CASE tools. He has published in Journal of Database Management, Journal of MIS, Data Base, Information Systems Management, Information and Management, and Communications of the ACM.

Seon H. Cho teaches strategic information systems at Korea Advanced Institute of Science and Technology. He received a Ph. D. in Computer Science from the University of Pittsburgh in 1963. Dr. Cho was the original member to implement the Wang Software Engineering Institute. Currently, he is also a chairman of the Wang Computer Korea. His main research interests include strategic IS and business process reengineering. Recently, he published in Journal of Database Management, Computer World and HiTech Information.