A Multimedia Information System Integrating Hypermedia, Information Retrieval, and DBMS Technologies *

Woong-Kee Loh, Byung-Kwon Park, Sang-Ho Lee, and Kyu-Young Whang
Computer Science Department and Center for Artificial Intelligence Research
Korea Advanced Institute of Science and Technology

Abstract: In this paper, we describe the design and implementation of a multimedia information system integrating hypermedia, information retrieval, and DBMS technologies. We browse the information space through hypermedia link navigation and also use ad hoc queries for direct retrieval of information from the database. Both the hypermedia and the information retrieval systems are integrated into a database management system (DBMS) so that concurrent edition, update, and retrieval of documents can have the ACID properties of atomicity, consistency, isolation, and durability. We will first explain the architecture of the system and then, describe how the hypermedia system is integrated with information retrieval capability, and finally integrated with DBMS capabilities.

1 Introduction

Due to advances in information technology and information superhighways, we live in a society flooded with multimedia information. The types and amount of available information have been increasing dramatically. Handling vast amount of information poses enumerous problems. Among them, we address the following ones: efficient storage and retrieval of multimedia information and organization of related information.

Hypertext or hypermedia has long been studied as an efficient tool for structuring, managing, and retrieving related information[Mar88]. *Hypertext* is nonlinear text that can be navigated among related contents freely with no predefined sequences[Con87]. In contrast, plain texts or books are read only sequentially. *Hypermedia* is an extension of hypertext[Nil90] consisting of not only text, but also multimedia data such as graphics, image, audio, and video.

As the information base grows large, navigation capability alone is not adequate for getting the information we want because every node in the path to the goal should be visited. Sometimes, it becomes difficult to decide where we are or where we should go next to get to the node that contains the information. Jakob Nielsen defined it as the *disorientation* problem, one of the major problems in hypermedia systems[Nil90].

Information retrieval has long been studied as another kind of information management technique. An information retrieval system automatically extracts keywords and builds an index using them. Documents are retrieved through queries composed of the keywords from text documents. Information retrieval is a key element in digital libraries and personal digital assistants. The information retrieval systems, however, have a problem that users have difficulties in expressing as a query the information they want[Sal88]. In addition, the keywords extracted from a document are not complete[Sal88]. As most of current hypermedia systems and information retrieval systems use file systems, they suffer from poor performance due to sequencial access to files for navigation or keyword search. Using the database systems, however, not only the performance can be enhanced by the indexing and query facilities, but also the stability of data is guaranteed by the concurrency control and crash recovery.

In this paper, we propose a multimedia information system that allows users to search the database via keyword-based queries as well as to navigate via hypermedia links. In the proposed system, the hypermedia and information retrieval systems are integrated with a database management system to provide efficient storage, management, and retrieval of large multimedia information.

This paper is organized as follows. In Section 2, we briefly describe the global architecture of the proposed system. In Section 3, we describe the integrated user environment of the system. In Section 4, we present integration of the hypermedia system and the database management system. In Section 5, we present integration of the information retrieval system and the database management system. Finally, we conclude the paper in Section 6.

2 Global Architecture of The Multimedia Information System

The multimedia information system that we propose has an architecture as shown in Figure 1. It consists of the multimedia database system with a storage manager, hypermedia browsing and authoring system, and information retrieval system. In this section, we describe the requirements of these components.

*This research was supported by Samsung Group under the project of Fundamental Research on Multimedia Engines.

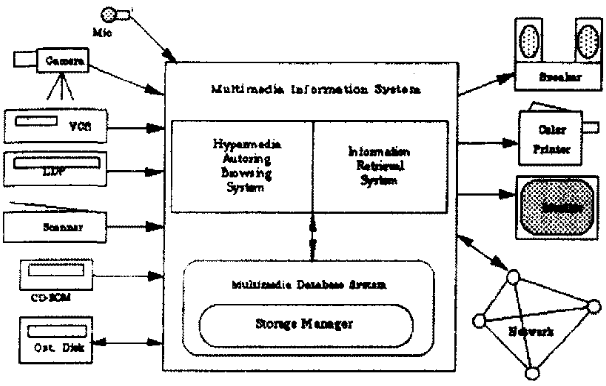

Figure 1. Multimedia Information System Architecture.

2.1 The Multimedia DBMS

To support multimedia applications, a multimedia DBMS should be added to handle large variable size objects efficiently. Variable size objects often involve partial insertions or deletions. These operations are necessary for editing linear data such as text, audio, and video.

A multimedia DBMS also needs new capabilities for simultaneous retrieval and updates of multimedia information by multiple users. Transactions in multimedia applications tend to be long-lasting and cooperative. Therefore, multimedia DBMS needs to develop a new concept of transactions.

In multimedia applications, the amount of data may exceed the capacity of a single storage device. Therefore, a multimedia DBMS should be able to distribute data in multiple storage media of possibly different types such as magnetic disks, magneto-optical disks, CD-ROMs, WORMs, magnetic tapes, and jukeboxes. These different types of storage devices form a storage hierarchy[Wal94], with fast, low-capacity devices at the top and with slow, high-capacity ones at the bottom.

A multimedia DBMS should be able to distribute data at the appropriate level in the storage hierarchy. It also needs to be able to easily adopt new devices as it evolves. A multimedia DBMS needs to handle new problems arising from this heterogeneity of storage devices. For example, in the case of query processing, traditional cost models and access methods based on magnetic disks would not be directly applicable in these environments[Ber88].

Multimedia applications frequently require mixed presentation of two or more media of different types. Time-dependent data such as audio, video, and animation must be synchronized to meet the presentation order, time, and duration specified by the user. For the synchronization of multimedia data, various parameters from the network environments, system platforms, and storage devices should be considered.

For the exact presentation of time-dependent data, the models for representing various forms of synchronization information need to be established. Recently, a synchronization model incorporating an augmented Petri-Net has been proposed[Lit90]. Also, the Amsterdam Hypermedia Model[Har94] has been proposed, extending the Dexter Hypertext Reference Model[Hal90] to deal with temporal aspects.

2.2 The Hypermedia System

In contrast to traditional sequential text, hypertext is nonsequential; there is no single order that determines the sequence in which the text is to be read[Nil90]. Hypertext consists of pieces of information called nodes interconnected through machine-supported links. Hypermedia is hypertext containing non-textual information such as images, audio, video, and animation. The concept of hypertext and hypermedia has been developed in the 1940's and has been studied extensively from the 1960's. Currently, a large number of hypertext and hypermedia systems exist including NLS/Augment, Guide, HyperTIES, NoteCards, Intermedia, and SEPIA[Con87].

One problem with existing hypermedia systems is that they are not suitable for handling a large amount of information. The simple model of nodes and links makes it difficult to find the information the user wants in a large hypermedia information space[Con87]. Therefore, search capability is essential for accessing a large hypermedia information space.

Searching in a hypermedia can be classified into content search and structure search[Hal88]. Content search involves searching the contents of all nodes and links. This can be carried out using the information retrieval techniques discussed in Section 2.3. Structure search specifically examines the hypermedia structure for subnetworks that match a given pattern. To support structure search, a query language for describing hypermedia network structures need to be designed, and efficient searching mechanism should be explored.

2.3 The Information Retrieval System

Information retrieval is concerned with locating specific information from a large collection of information in response to the users' requests[Sal88]. Much of the research on multimedia information retrieval has focused on content-based retrieval on text. Recently, there has been growing interest in content-based retrieval on image, voice, and video[Gro94].

There are representative index structures for text information retrieval systems: the inverted index file structure and signature file structure. The inverted index has been widely used in the text information retrieval system, and is known to be useful for the applications where update on text is rare. In contrast, signature is known to be suitable for the applications where update on text is frequent. An appropriate index structure should be used according to the target applications.

3 Integrated User Environment of The Multimedia Information System

The integrated user environment of the proposed multimedia information system provides a browsing capability for the hypermedia and the query capability for information retrieval as shown in Figure 2. The integrated environment

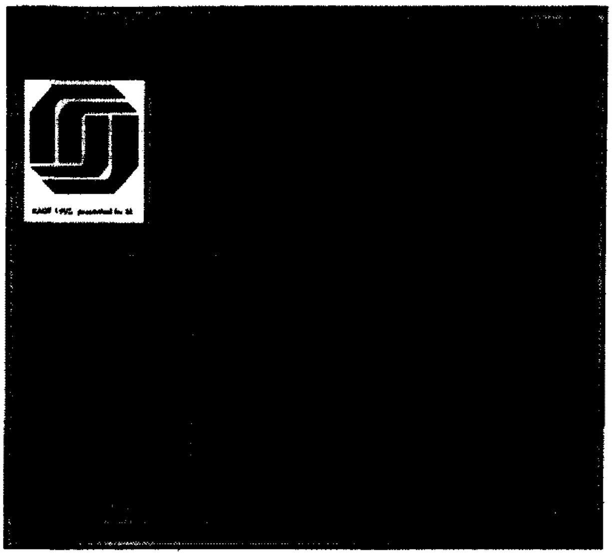

Figure 2. User Environment of Multimedia Information System.

enbles users to retrieve information by the hypermedia link navigation or by the keyword-based queries. Navigation is an easy and convenient access method. Navigation is adequate in a small information space. For a large information space, however, navigation only has difficulties locating the desired information. To solve this problem, information retrieval techniques should be integrated into hypermedia systems.

A typical process of obtaining information from the integrated environment is as follows. The user first inputs a query describing the desired information. Then, the names of all the relevant documents are shown to the user. As the user selects one of them, the selected document is opened and presented. The user can then start navigation from the document presented through hyperlinks.

Information retrieval via keyword-based queries reduces the cognative overhead and shortens the access paths to the target information. In particular, query operations can be used as an information filter of hypermedia systems.

4 Integration of The Hypermedia System and The Database System

The hypermedia system of our multimedia information system is designed based on the Dexter Reference Model. It contains the Within-Component Layer and the Storage Layer. In this Section, we explain how the objects of each layer of the hypermedia system are stored in the database system.

4.1 Within-Component Layer

The data structure and the database schema of the Within-Component layer are organized according to the MHEG definitions as shown in Figure 3[Wha96]. The DB_Content class, which corresponds to the MHEG Content class, has an attribute of the type DB_Content_Data class, which corresponds to the type MHEG Content-Data class, for storing presentation media content. The DB_Content_Data class has an attribute of the type DB_Included_Content class, which corresponds to the type MHEG Included-Content class, for storing raw data. As an extension of MHEG data structures, we define several subclasses of the DB_Included_Content class for storing raw data of their own specific type: they are DB_Line, DB_Rectangle, DB_Ellipse, DB_Text, DB_Digital_Video, DB_Audio, and DB_Image classes.

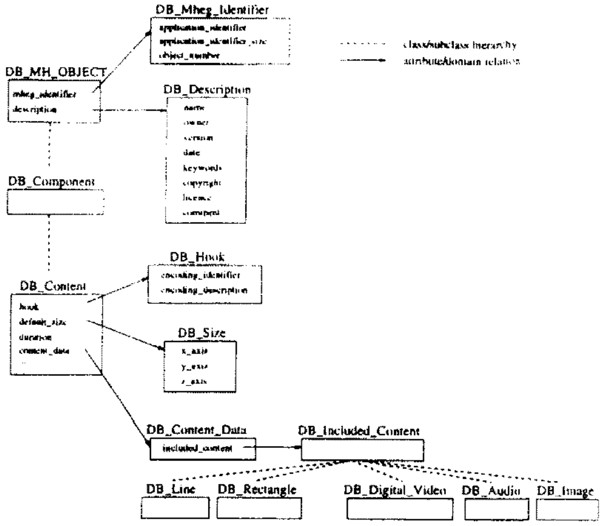

Figure 3. Storing the Hypermedia Data in the Database.

4.2 Storage Layer

In the Storage layer, we have the atomic component and the link component corresponding to hypermedia nodes and links, respectively. The atomic component in the Dexter model is defined using the data structure of the MHEG Composite class to support synchronization. An MHEG Composite object contains attributes defining trigger conditions and actions for synchronization and an attribute pointing to the MHEG Content objects list. As the CC_AtomicComp class has the same structure as the MHEG Composite class, atomic component objects can be easily encoded according to the MHEG standard for data interchange.

The link component of the Dexter model has the same data structure as the MHEG Link class. Its C++ interface class is the CC_LinkComp class and its database class is the DB_LinkComp class. Figure 4 shows the structures and the inter-relationships of the CC_LinkComp class and the DB_LinkComp class. As in the case of the atomic component, link component objects can be easily encoded according to the MHEG standard for data interchange.

The link components are necessary for hyperlink navigation. By storing link components in the database system, we can use efficient search capability of the database system for hyperlink navigation. That is, we can efficiently find the destination objects of DB_LinkComp objects that have the source object specifier value for the trigger_condition attribute.

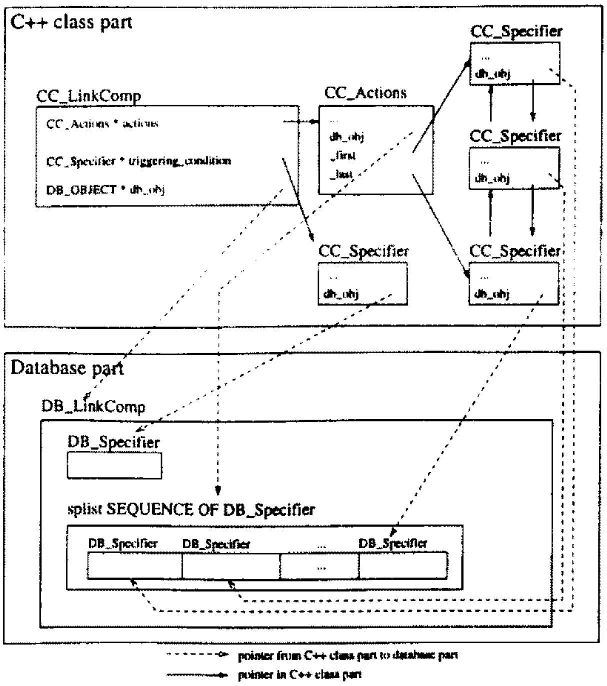

Figure 4. Relationships between CC_LinkComp and DB_LinkComp.

The rationale that we can define the link component data structure of the Dexter Model according to the MHEG definitions is that both models define the hyper links independently of the node data structure. A typical contrary example is the World-Wide Web (WWW) Hypertext Markup Language (HTML) [Mac94] documents containing every hyperlink within themselves. This structure prohibits efficient search based on the various properties of the links.

The data structure defining hyperlinks independently of the nodes enables us to use indexing capabilities of the database system to perform efficiently operations on hyperlinks. That is, by generating indexes on the OIDs of hyperlink source and destination objects and other attributes, efficient retrieval of hyperlinks and navigation can be provided. The operations on hyperlinks include the operations of finding destination objects from specific source objects and finding every hyperlink from or to a certain node. We can also use SQL query language on hyperlinks.

5 Integration of the Information Retrieval System and the Database System

Integration of the information retrieval(IR) system and the database system can be categorized into two approaches: loosely coupled and tightly coupled. The loosely coupled approach stores text data in the database, but maintains the index of the text in external files and uses the index files for query processing. The approach can easily integrate information retrieval systems and database systems, and use various fast index file structures provided by the IR system. In general, however, the approach makes it difficult to support consistency of text data in the database and external

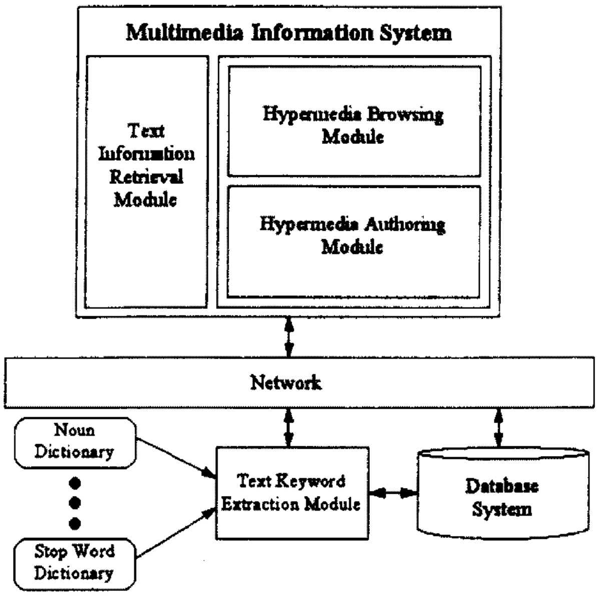

Figure 5. Integration of the Information Retrieval System and the Database System.

index files, and to provide concurrency control and recovery capabilities in multi-user environments. In contrast, the tightly coupled approach maintains index structures in the database and let the database system manage the text and index data. The approach keeps ACID properties - atomicity, consistency, isolation, and durability - on text and index data.

The tightly coupled approach is divied into two methods: one is to integrate text indexes with a database system, and the other to integrate with the storage system. In this Section, we explain each of the two tightly coupled integration methods.

5.1 Integration with the Database System

This integration method defines a text index schema and stores text indexes using the schema through the interfaces provided by the database system. The text index schema includes a module to extract keywords from the input text as well as the attributes. Figure 5 shows the architecture of the integrated information retrieval and database system according to this method.

The procedures in the integrated system to store the extracted keywords into the database and to process user queries using the indexes are as follows. In the text index generation phase, the system first stores the text data into the database and invokes the keyword extraction module as specified in the index schema with the input of the text data. The keyword extraction module stores keywords extracted from the text data as instances of the index schema. In the query processing phase, the system first analyzes the user query, a Boolean expression consisting of keywords and logical connectives, and executes the SQL query statement converted from the Boolean expression.

In this method, we do not need to extend the database

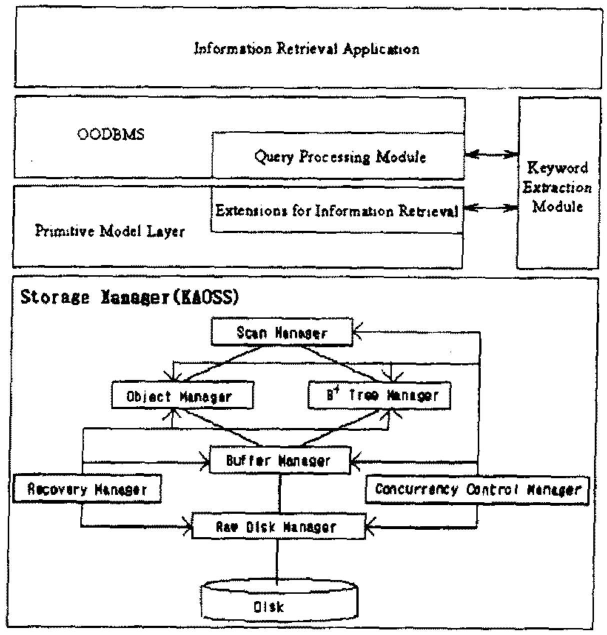

Figure 6. Integration of Information Retrieval System and Storage System.

system for information retrieval. We also can execute text operations - editing, deleting, indexing, and retrieving - under the control of the database management system. Therefore, this method keeps every advantage of database systems such as concurrency control, recovery, consistency, etc. However, this method suffers from performance degradation as the number of documents and keywords increases[Sam95].

5.2 Integration with the Storage System

This method stores text indexes into the storage system. Figure 6 shows an architecture of the storage system extended for information retrieval.

In this method, the procedures for storing the extracted keywords into the storage system and for processing user queries using the indexes are as follows. In the text index storage phase, the user invokes the keyword extraction module passing the user text as the input, and stores the user text and extracted keywords into the storage system using its interface functions. In the query processing phase, the system first analyzes the user query, a Boolean expression consisting of the keywords of the user query and connectives among the keywords, and then, retrieves the texts which satisfy the Boolean expression.

This integration method has difficulties in extending the existing storage system. However, it has the advantage of providing scalable performance to the index size[Sam95] as well as all the database system capabilities. The multimedia information system of this paper has integrated information retrieval system by extending a storage system, KAOSS (KAIST Object Storage System), developed in the Database and Multimedia Laboratory at KAIST.

6 Conclusions

Research and development on hypermedia systems have been extensively performed since 1940's. However, as the size of the information space and the number of nodes and links grow large, navigation only is not adequate for retrieving the desired information from the multimedia database. Besides, due to the disorientation problem, it is difficult to determine where I am and where I should go to get to the piece of information that I want.

To solve this problem, in this paper, we have proposed a multimedia information system that integrates the hypermedia system with the information retrieval system and provides an environment in which users can exploit navigation via hyperlinks and search via keyword-based queries. The multimedia information system also integrates the hypermedia system and the information retrieval system with the database system so as to provide the efficient storage, management, and retrieval of large multimedia information.

In this paper, we have described the overal architecture of the multimedia information system, the integrated user environment of hypermedia and information retrieval systems, the integration of the hypermedia system with the database system, and finally integration of the information retrieval system with the database system. We are currently developing efficient storage and retrieval methods for structured multimedia documents such as HTML and HyTime used in the applications of the World Wide Web and digital libraries. The multimedia information system we have described serves as the research platform.

References

[Aks88] Akscyn, R., McCracken, D., and Yoder, E., "KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations," *Communications of the ACM*, Vol. 31, No. 7, pp. 820-835, July 1988.

[App87] Apple Computer Corp., HyperCard User's Manual, Cupertino, Calif., 1987.

[Bai88] Baird, P. and Percival, M., "Glasgow Online: Database development using Apple's HyperCard," In *Proc. Online Information*, London, pp. 207-218, Dec. 1988.

[Ber88] Bertino, E., Rabitti, F., and Gibbs, S., "Query Processing in a Multimedia Document System," In *ACM Transactions on Office Information Systems*, Vol. 6, No. 1, Jan. 1988.

[Bus45] Bush, V., "As We May Think," *Atlantic Monthly*, pp. 101-108, July 1945.

[Con87] Conklin, J., "Hypertext: An Introduction and Survey," *IEEE Computer*, Vol. 20, No. 9, pp. 17-41, Sept. 1987.

[Del86] Delisle, N. and Schwartz, M., "Neptune: A Hypertext System for CAD Applications," In *Proc. Intl. Conf. on Management of Data*, ACM SIGMOD, Washington, D.C., pp. 132-143, May 1986.

[Eng63] Engelbart, D.C., "A Conceptual Framework for the Augmentation of Man's Intellect," In *Vistas in Information Handling*, Vol. 1, Spartan Books, London, 1963.

[For84] Forsdick, H. C., Thomas, R. H., Robertson, G. G., and Travers, V. M., "Initial Experience with Multimedia Documents in Diamond," In *Proc. IFIP 6.5 Working Conf. Computer Message Services*, IFIP, 1984.

[Fur90] Furuta, R. and Stotts, P. D., "The Trellis Hypertext Reference Model," In *Proc. Hypertext Standardization Workshop*, National Institute of Standards and Technology, pp. 83-94, Jan. 1990.

[Gro94] Grosky, W. I., "Multimedia Information Systems," *IEEE Multimedia*, Vol. 1, No. 1, Spring 1994.

[Hal88] Halasz, F., "Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems," *Communications of the ACM*, Vol. 31, No. 7, pp. 836-852, July 1988.

[Hal90] Halasz, F. and Schwartz, M., "The Dexter Hypertext Reference Model," In *Proc. Hypertext Standardization Workshop*, National Institute of Standards and Technology, pp. 95-134, Jan. 1990.

[Har94] Hardman, L., Bulterman, D. C. A., and Rossum, G. V., "The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model," *Communications of the ACM*, Feb. 1994.

[ISO94] ISO/IEC JTC1/SC29/WG12, MHEG Draft Internation Standard, Oct. 1994.

[Leg88] Legget, J., Schnase, J. L., and Kacmar, C. J., Working Definition of Hypertext, TAMU 88-020, Hypertext Research Lab., Texas A&M Univ., 1988.

[Leg90] Leggett, J. J., "Reference and Data Model Group: Comparison of Three Models," In *Proc. Hypertext Standardization Workshop*, National Institute of Standards and Technology, pp. 15-18, Jan. 1990.

[Lit90] Little, T.D.C. and Ghafoor, A., "Synchronization and Storage Models for Multimedia Objects," In *IEEE Journal on Selected Areas in Comm.*, Vol. 8, No. 3, Apr. 1990.

[Mac94] McArthur, Douglas C., "World Wide Web & HTML," *Dr. Dobb's Journal*, pp. 18-26, Dec. 1994.

[Mar88] Marchionini, G. and Schneiderman, B., "Finding Facts vs. Browsing Knowledge in Hypertext Systems," *IEEE Computer*, Vol. 21, No. 1, pp. 70-79, Jan. 1988.

[Mey86] Meyrowitz, N., "Intermedia: The Architecture and Construction of an Object-Oriented Hypermedia System and Applications Framework," In *Proc. ACM Conf. on Object-Oriented Programming, Systems, Languages, and Applications*, pp. 186-201, Sept. 1986.

[Mey88] Meyrowitz, N., A Short Course on Hypertext, Hypertext Research Lab., Texas A&M Univ., 1988.

[Nil90] Nielson, J., *Hypertext and Hypermedia*, Academic Press, Inc., 1990.

[Sal88] Gerard Salton, *Automatic Text Processing*, Addison-Wesley Publication Co., 1988.

[Sam95] Samuel DeFazio et al., "Integrating IR and RDBMS Using Cooperative Indexing," In *Proc. ACM SIGIR*, pp. 84-92, 1995.

[Shi91] Shimojo, S., et al., "A New Hyperobject System Harmony: Its Design and Implementation," In *Proc. Intl. Conf. on Multimedia Information Systems*, pp. 243-257, Nov. 1991.

[Shn87] Shneiderman, B., "User Interface Design for the HyperTIES Electronic Encyclopedia," In *Proc. Hypertext '87 Conf.*, Chapel Hill, pp. 189-194, Nov., 1987.

[Str89] Streitz, N. A., Hannenmann, J., and Thuring, M., "From Ideas and Arguments to Hyperdocuments: Travelling Through Activity Spaces," In *Proc. Hypertext '89 Conf.*, Pittsburgh, pp. 343-364, Nov. 1989.

[Wal94] Wallace, S., "Managing Mass Storage," In *BYTE*, Vol. 19, No. 3, Mar. 1994.

[Wha96] Kyu-Young Whang et al., "An Object-Oriented Hypermedia System Based on the Dexter Reference Model and the MHEG Standard," *IEIEC Trans. on Information and Systems*, 1996.