CHAPTER 3

DATABASE ISSUES

1. Introduction

Most current hypertext systems use proprietary data formats for storage. Some researchers have tried using Relational Data Base Management Systems (RDBMS) to implement hypertext applications [Schutt & Streitz, 1990], [Meyrowitz, 1986]. There exists a large class of applications for which relational database management systems and other more conventionally structured database systems are too limited. These applications can be characterized as complex, large−scale, data−intensive programs such as CAD/CAM systems, documentation management systems, hypermedia systems, and geographical information systems. This class of applications needs a database model that is more expressive and flexible than the relational model. Some researchers believe that object−oriented databases can meet the data handling requirements of such applications.

Among the many research issues in the area of hypermedia, database requirements of hypermedia systems have received very little attention. HyperBase, based on an object−oriented data model, was implemented on Sybase, a RDBMS. Similarly, Intermedia was originally implemented on INGRES and later on C−Tree. Very little work has been done in the usage of Object−Oriented Data Base Management Systems (OODBMS) for hypermedia applications. The use of an OODBMS for a hypermedia system is an excellent marriage between the rigorous infrastructure and modeling abilities of object−orientation and the flexibility and navigational access to information characterized by hypermedia [Lange, 1993].

2. Object−Oriented Concepts and Hypermedia

Before we discuss the database requirements of hypermedia, let us try to understand the object−oriented paradigm and analyze its relevance to hypermedia. The advantages of using an OODBMS includes the ability to model complexity, to provide a unified representation of the problem domain, and to build complex objects.

There have been object−oriented approaches to systems analysis and design, programming languages, operating systems and database systems. Object−oriented systems provide many new features that are not present in conventional structured systems. The object−oriented paradigm more closely represents the problem domain by mapping abstractions of real world entities as objects/classes. It also focuses on the definition and inheritance of behavioral capabilities in the form of operations embedded within objects. The paradigm also supports simpler capabilities for structuring complex objects.

Object−oriented systems, including languages and database management systems support concepts such as objects, classes, methods, data abstraction, encapsulation, inheritance, schema evolution, reusability,concurrency control, correctness, polymorphism, composite objects, active objects, version control and direct manipulation [Cattell, 1991].

Many features from the world of object−oriented analysis, design and programming can be extended to the hypertext model.

1. Simple nodes can be compared to atomic objects representing primitive data types such as integer, character, string, video frame and bitmap.

2. Objects or nodes can be accessed using object identifiers or node identifiers.

3. A link can be represented by a set of at least two object identifiers. Links can also be treated as objects with their own identifiers (link identifiers) which can be used to separate index information from content.

4. A composite node in hypertext (made of webs of nodes and links) can be treated as a composite object or an aggregation of simple objects.

5. The concepts of data abstraction and encapsulation can be applied by defining methods to create, delete, update and manipulate nodes and links, to traverse links, and to trigger events.

6. Nodes and links can be grouped under different classes based on structural and behavioral patterns (semantics). Organizing nodes and links semantically helps manage the network better, eliminates ambiguity, clearly differentiating the purposes of these objects.

7. Nodes and links of a particular class can also inherit properties from related superclasses. This feature can be used in the creation and management of hypermedia templates – when a user changes a parent template, these changes can be propagated to all its sub–classed templates.

8. Whenever the properties of nodes and links have to be changed it should be easy to do so through schema evolution.

9. Other object–oriented concepts that are pertinent to hypertext include concurrency control, versioning, and persistence.

In HyperBase, the basic class was a HB_Object which had three subclasses: HB_Node, HB_Link, and HB_Composite_Object [Schutt & Streitz, 1990]. HB_Nodes are HB_Objects with content and history. HB_Links are HB_Objects that connect two existing HB_Objects. HB_Composite_Objects are collections of references to existing HB_Objects. The generic operations of create, modify, copy, delete, retrieve an object (part of or whole) were supported for nodes, links, composite objects, and attributes. Since the data model was object–oriented, a natural choice to implement HyperBase would have been an OODBMS. However, for reasons of availability, a first prototype was implemented on Sybase, a RDBMS.

3. Requirements for Next–Generation Hypermedia Systems

Frank Halasz had originally identified seven issues for next generation hypermedia systems [Halasz, 1988]. Other researchers have come up with a few more over the past few years [Lange, 1993], [Fountain et al., 1990]. Object–oriented database management systems seem to meet most of these requirements.

a. Openness and Distribution

Most current hypertext systems are basically closed systems. That is, material created in one system cannot be easily integrated with material created in another system. An open hypermedia system is a system which can connect to other information systems (both hypertext and non–hypertext). It should be able to freely exchange data. Distribution allows the system to store at geographically dispersed sites in a manner transparent to the user.

b. Support for collaborative work or sharing

Future generation hypermedia systems must provide adequate support for collaborative work. This includes simultaneous multi–user access to the hypermedia network, robust concurrency control mechanisms, broadcasting to users any changes made to the network by other users, and tracking contributions made by each member of a team.

c. Data integrity/Correctness

The database layer should preserve data integrity and provide traditional secondary storage management and data administration facilities. It should forbid dangling references to objects or attributes and prevent recursive inclusion of composite objects [Schutt & Streitz, 1990]. It should also support either the notion of rules as in extended relational databases or the concept of semantics as in object−oriented databases.

d. Dynamism

Most current generation hypermedia systems deal with static information. Future systems need the ability to dynamically reconfigure the network in response to changes made to the network or its contents. Virtual or dynamic structures (active objects) are similar to dynamically generated views in relational databases. Virtual structures are possible only when current models are changed to adapt to changing information.

e. Search and query mechanism

In addition to navigational access to information, hypermedia systems should provide efficient search and query mechanisms. This would partly solve the "lost in space" (disorientation) problem experienced by users during navigation. There can be two kinds of queries − a structure query to retrieve a part of the network and a content query to retrieve a specific node. The former requires the development of a query language geared towards dealing with hypertext network structures.

f. Computation

The integration of AI into hypermedia engines will be an interesting area to explore. This would include inheritance, truth maintenance, rule based searching, and inference engines. Computation built into the hypermedia system is likely to be more efficient for extensive access to information in the network.

g. Composites

With increasing use of hypermedia for sophisticated applications it becomes necessary to deal with groups of nodes and links. This will require making a composite node as a primitive construct in the basic hypermedia model. Inclusion (or part−of) relations have to be supported in addition to standard, referential links.

h. Versioning

Versioning is required in order to keep track of changes to the network. This can include versioning at the level of individual entities such as nodes and links and also at the level of the hypermedia network as a whole.

i. Multimedia Support

The database layer should be able to efficiently store and retrieve multiple media. It should also provide transparent access to different storage media.

j. Extensibility and Tailorability

Extensibility involves the ability to handle extensions to the existing data model (schema evolution) in a flexible and safe manner. The database layer should not only be able to handle the structural part of the hypertext data model but also the semantic part to ensure data abstraction and encapsulation of the evolving data model.

4. Object−Oriented Databases and Hypermedia

Intermedia, developed at Brown University, is an object−oriented hypermedia system implemented on a relational database management system, INGRES. In an experimental study conducted by researchers at the University, INGRES was compared with an object−oriented database management system (Encore, developed in–house) for the implementation of Intermedia [Smith & Zdonik, 1987].

RDBMS	OODBMS
Hierarchies to be flattened into relations	Hierarchies can be represented as they are.
Objects retrieved through query language	Operations on objects stored in the database
Many queries to retrieve a single object	One message sent to an object which can send messages to other objects
Results from queries to be stored in application's data structure	Messages can directly manipulate data in the database and return a pointer to the result
High level of data structure and impedance mismatch	Lower level of impedance and data structure mismatch
Integrity checks enforced only during commit time	Integrity checks performed during run-time through triggers
Due to the flat nature of relations each record in each relation in a given hierarchy to be explicitly locked	An entire hierarchy can be locked in a single operation

Table 3.1 – Results of Evaluation at Brown University [Smith & Zdonik, 1987].

At the conclusion of the study, researchers felt that an object–oriented database could successfully meet the data handling requirements of Intermedia.

In a more recent study conducted by researchers at the University of Tokyo, four different types of storage mechanisms were evaluated against some of the requirements listed in Section 2. The following table below shows the results of the evaluation.

	Filesystem	KnowledgeBase	RDBMS	OODBMS
Openness	Partly	No	Yes	Yes
Sharing	Partly	No	Partly	Yes
Integrity	No	Yes	Yes	Yes
Multimedia	No	No	No	Yes
Querying	Partly	Yes	Yes	Partly
Versioning	Partly	No	No	Partly
Extensibility	No	Yes	Yes	Yes

Table 3.2 – Results of Evaluation at University of Tokyo [Lange, 1993].

From Table 3.2, it is apparent that an OODBMS scores high on all aspects of the reqirements for a storage mechanism for hypermedia systems. Without going into details on the file system approach and the knwledge base approach, let us compare the RDBMS approach against the OODBMS approach (Table 3.3 below).

RDBMS	OODBMS
Easy to implement since concepts are well known	Technology is new and hence difficult to implement
Bad performance even for simple link traversals	Navigation model of the database can be directly exploited
Object-identity and composite objects to be directly modeled in the hypertext model rather than the database	Commonalities between the database and the hypertext model facilitate direct implementation of data structures and behavior
Complex data to be flattened leading to loss of data abstraction, encapsulation and performance	Efficient data handling mechanisms for complex data
Not very good support for collaborative work such as version management and concurrency control	Flexible transaction processing facilities and wide range of locking policies

Table 3.3. – Specific Differences between an RDBMS and an OODBMS [Lange, 1993].

As an outcome of their experimental studies, this research team is investigating into an object−oriented extension to the hypertext data model. The cross−breeding of object−orientation and hypertext is called object−oriented hypermodeling. This concept has been utilized in a tool for rapid object−oriented application development on an object−oriented database. The tool supports the hypermodeling process through a graphical user interface, a C++ class library and Object SQL on the object−oriented database ONTOS DB. The combination of object−oriented hypermodeling and ONTOS DB together satisfied all the requirements listed in Section 2 except version control [Lange, 1993]. Some recently released OODBMS such as Versant, Statice, and O2 have been used in developing hypermedia applications [English, 1992].

In summary, these studies have shown that object−oriented database management systems are more appropriate as storage mechanisms for hypermedia systems than traditional methods. Researchers are also working on a new class of systems called hyperbases. A hyperbase is a storage mechanism with a built−in hypertext data model providing primitives to manipulate instances of this model. It has also been proposed that by integrating an object−oriented data model with hypertext, the information retrieval process can be greatly improved [Rao et al., 1993]. This can be accomplished by redirecting hypertext functionality from the application level to the database level using object−oriented database systems.

5. Summary

Object−oriented technology provides the ability to model, to develop and to manage complex systems that cannot be easily implemented with current technology. Hypermedia systems constitute an emerging class of complex management information systems. They should facilitate distributed memory, distributed processing, and distributed knowledge on a network. An object oriented approach to hypermedia may help us achieve this goal much better than current approaches. However, research in this area is still in its infancy and there are many open issues to be investigated. Any breakthroughs in this arena will take us closer to realizing Ted Nelson’s goal, "Everything should be available to everyone. Any user should be able to follow origins and links of material across boundaries of documents, servers, networks, and individual implementations. There should be a unified environment available to everyone providing access to this whole docuverse." [Nelson, 1991].

References

[Cattell, 1991]. Cattell, R.G.G. What are Next−Generation Database Systems ? , CACM, October 1991.

[English, 1992]. English, Larry. Object Databases at Work, DBMS, October 1992.

[Fountain et al., 1990]. Fountain, Andrew M., Hall, Wendy, Heath, Ian and Davis, Hugh C. MICROCOSM: An Open Model For Hypermedia With Dynamic Linking, Proceedings of ECHT ’90, 1990.

[Halasz, 1988]. Halasz, Frank. Reflections on NoteCards: Seven Issues for Next Generation Hypermedia Systems, CACM, July 1988.

[Lange, 1993]. Lange, Danny. Object−Oriented Hypermodeling of Hypertext Supported Information Systems, Proceedings of HICSS ’93.

[Meyrowitz, 1986]. Meyrowitz, Norman K., Intermedia: The Architecture and Construction of an Object−Oriented Hypermedia System and Applications Framework, OOPSLA ’86 Proceedings.

[Nelson, 1965]. Nelson, Ted. A File Structure for The Complex, The Changing and The Indeterminate, ACM 20th National Conference, 1965.

[Nelson, 1991]. Nelson, Ted. Speech delivered at MultiMedia Expo ’91, New York.

[Rao et al., 1993]. Rao, Gururajan R., Balasubramanian, V., and Suresh, B.S. Integration of Hypertext and Object−Oriented Databases for Information Retrieval, Proceedings of the Nineteenth Annual Northeast IEEE Conference on Bio−Engineering, IEEE Press, March 1993.

[Schutt & Streitz, 1990]. Schutt, Helge A., and Streitz, Norbert A. HyperBase: A Hypermedia Engine Based on a Relational Data Base Management System, Proceedings of ECHT '90, 1990.

[Smith & Zdonik, 1987]. Smith, Karen and Zdonik, Stanley. Intermedia: A Case Study of the Differences Between Relational and Object−Oriented Database Systems, OOPSLA '87 Proceedings.