CMU-ITC-89-073

Seoul Korea April 1989

International Symposium on Database Systems for Advanced Applications

On Using ODA for Multimedia Database Applications

Mark Sherman[1]
Ann Marks
Jonathan Rosenberg
Michael Zheng[2]
Jaap Akkerhuis

Information Technology Center
Carnegie Mellon University
Pittsburgh, PA 15213
USA

Abstract

This paper provides some background on multimedia systems and ODA, and considers whether ODA can meet some needs of multimedia database systems: multiple kinds of primitive data, multiple presentation of data, multiple description of data, extensibility of primitive data, multiple structures imposed on data, limited repetition of structure, efficient representation of data. Features and shortcomings of ODA for supporting multimedia database systems are discussed.

I. Introduction

The EXPRES (EXPerimental Research in Electronic Submission) project at Carnegie Mellon University was funded by the US National Science Foundation to promote the electronic interchange of multimedia documents among the scientific research community. To support electronic interchange of documents, we found it necessary to create, manipulate, extract, combine, archive and retrieve multimedia documents created by different systems using different formats. One of our initial tasks has been to investigate how to exchange multimedia documents between heterogeneous systems. We decided to use the new International Standard "Office Document Architecture and Interchange Format" (ODA) [ISO 9541] as the basis for our exchange facility [Rosenberg]. However, it is clear that another task is to provide a way to integrate together multimedia documents that are created during collaborative work: electronic mail messages between investigators, diagrams and illustrations of experiments, derivations of the mathematics being used to support an experiment, tables of data generated by preliminary experiments, excerpts from previous reports, and so on. Therefore, we are faced with a need for a multimedia database system. Given that we are using ODA as our exchange medium, the question arises "Is ODA appropriate as a database representation?" This paper considers the requirements of a multimedia database system and evaluates ODA against those requirements.

The paper is organized as follows: Section II discusses the features and implementations of multimedia systems in general. In section III, we isolate specific features of multimedia documents that would be used in database applications. Section IV provides a brief explanation how ODA represents multimedia information. In section V, we evaluate the features provided by ODA against the needs for database systems. The last section summarizes our thoughts on the use of ODA for multimedia database applications.

[1]This work was funded by the US National Science Foundation under grant ASC-8617695. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation or the US Government.

[2]Current address: Department of Computer Science, Marquette University, Milwaukee, WI. 53233, USA.

II. Background on Multimedia Systems

For purposes of this paper, we consider a multimedia system as one that manipulates documents that contain multi-font text, raster images, geometric drawings, animations, equations, tables, graphs and sound. An example fragment of a multimedia document is shown below. It describes Pascal's Triangle with its equations, an animation showing how the triangle is built, and a spreadsheet implementing the algorithm.

There are a substantial number of such systems. Some experimental systems, such as the ARPA experimental multimedia mail system [Reynolds] and MINOS [Christodoulakis] have limited deployment. Others, such as MULTOS, are targeted for commercial use, but are in still in development [Bertino]. Some, such as the Diamond system from BBN [Thomas] are commercially available. The Andrew Toolkit and its associated applications [Palay, Borenstein] have a wide distribution as part of the MIT X Window System distribution. The Andrew system is used by several thousand users at CMU who create multimedia documents for classes, exchange them as mail, and post multimedia bulletin board messages. The Andrew system at CMU accepts one new bulletin board message every 20 seconds to one of over 1700 bulletin boards, accumulating nearly 10 gigabytes each year.

As described so far, the need for a multimedia database is clear. Searching through the mass of messages trying to locate some desired information or trying to correlate information between a collection of messages is exactly the province of databases. However, the problem is complicated by the need to manipulate documents created by different multimedia systems. For example, a goal of the EXPRES project is to allow collaboration amongs researchers using different systems. Another EXPRES grantee, the Center for Information Technology Integration at the University of Michigan, uses the Diamond multimedia system as the basis of its document system. Therefore, users at Carnegie Mellon University must be able to manipulate Diamond documents and users at the University of Michigan must be able to manipulate Andrew documents. Even within Carnegie Mellon University, a large number of documents are produced using the Macintosh and DOS machines with a variety of programs. Documents created outside of the Andrew system must be integrated into the Andrew system in order to participate fully in any multimedia database system.

Thus we decided that a common, but general format should be used to accomodate the need for heterogeneous systems to communicate. As necessary, systems would translate into and out of that common format. Instead of designing some private format for use only in the EXPRES project, we decided to search

Pascal's Triangle

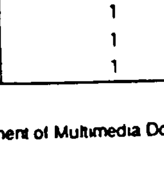
This table contains several descriptions of Pascal's Triangle. It contains a set of equations which defines the values of the triangle. It also contains an animation showing the building of the triangle. Finally there is an implementation of Pascal's Triangle using the spreadsheet facilities of the table object.

In order to run the animation, click into the cell and choose the animate item from the menus.

$v_{0,j} = v_{i,0} = 0$

$v_{1,1} = 1$

$v_{i,j} = v_{i-1,j} + v_{i,j-1}$



1	1	1
1	2	3
1	3	6
1	4	10
1	5	15

Figure 1: Fragment of Multimedia Document (from Andrew)

for a representation that might be supported by industry outside of the project as well. We decided to use the new ODA standard for importing and exporting documents. As we reported in [Rosenberg], the use of ODA for this purpose seems appropriate.

III. Use of Multimedia Information in Databases

We characterize the differences between conventional databases and multimedia databases according to the following criteria:

1. Kinds of data
2. Presentation of data
3. Description of data
4. Extensibility of primitive data
5. Structure imposed on data
6. Repetition of structure
7. Representation of data

Each of the characterizations is discussed below.

The first difference is that conventional and multimedia databasea contain different kinds of primitive data. A conventional database system contains limited kinds of primitive data, usually numbers and words. Some database systems permit specialization of the primitive data, such as defining some numbers as time and others as a currency value. However, the amount of information in each datum is relative small. A multimedia database has a much richer collection of primitive data, including several kinds of graphical data (raster, line drawing, graphs, charts, video) and several kinds of audio data (music, sound, voice). Therefore the database system must be able to accomodate ways to store a variety of primitive data. Further, the primitive operations available to the database user must permit manipulation of each primitive kind of data.

A second difference between a conventional and multimedia database is the number of presentations of primitive data. In a conventional database, a string or number usually has only one presentation: the characters in the string or digits in the number. However, a list of numbers in a multimedia database could have several presentations for the same data. For example, the list of numbers might be monthly sales for a particular region. However, the presentation in a document could be a line graph for the period, a table of numbers used in a spreadsheet-like medium, or a pie chart showing relative values. Many multimedia systems allow a presentation to change, and some, such as Andrew, permit multiple presentations of data to be simultaneously present in a document. Thus one might have to search not only for some primitive data, but for its presentation as well.

The use or lack of multiple presentations of data in a multimedia database is a third difference. In addition to multiple presentations, a multimedia database system will frequently provide multiple descriptions of some of its data. For example, a figure in a document will typically have both a caption and the actual picture. A request to find a particular picture might involve searching both the picture for a particular structure, for example, a "country" tag in a map, as well as the caption of a figure describing the map. A conventional database system only provides a limited kind of data description through the use of a data dictionary. The data dictionary describes all data using a certain format however, and not particular instances of that data.

A fourth difference is the extensibility of primitive data. Most database systems provide no way to extend the primitive data available, except for some simple renaming. For example, one might be able to specify a particular number to be a temperature and another to be an employee ID, and a good database system will even forbid adding temperatures and employee IDs. However, conventional database systems do not permit the user to define, say, a raster image as a new primitive datum. Multimedia systems are evolving towards allowing the users to create arbitrary kinds of new media. For example, the Andrew system has very few media as part of the initial toolkit, but users have created media for calendars, piano music, calculators, style sheet editors, various programming languages, and graph-theoretic networks. A multimedia database system would have to be able to cope with new kinds of media beyond its initial design.

A fifth difference is the amount of structure imposed on data. In general, multimedia systems provide for multiple, hierarchical relationships among their data. A relational system typically has only one structure for the data — a table. In contrast, each document can be structured in at least two ways: logically as a collection of chapters, each of which has sections, each of which has paragraphs (and so on), or physically as a sequence of pages that result from imaging the document. One needs to be able to carry out searches exploiting any of the relationships. For example, one might either search all bibliographies for a reference, or one might search all first pages for a particular picture. Although, one can represent any structure as a collection of tables, the resulting collection is not intuitive and difficult to search.

The sixth difference we discuss is that the amount of structure repetition in a multimedia database differs from the amount in a conventional database. For example, the description of a record in a data dictionary provides structure in a relational database. It is not unusual to have several thousand records (rows) in a table. Thus there is little structure (one row) but it highly repeated (many rows). Database systems exploit the repetition of structure in indexing and other optimizations for queries. However, the structure in a multimedia system may not be as repetitive. For example, a document may have fewer than 30 chapters, each of which has fewer than 20 sections. Further, different documents will have different structures. For example, one document that is a textbook will be organized around chapters, while another document for a homework assignment is organized around questions to be answered. Thus there is much less repetition of structure within a multimedia database, and less opportunity to optimize queries.

The final difference we consider is the representation of data in a database. Conventional database systems use simple encodings, such as ASCII. Usually there is hardware support for searching and comparing these representations. In contrast, multimedia documents usually have a complicated encoding for the structure of the document, and multiple representations for the primitive data. For example, the separation of a document into chapters is done by special codes within the document itself. The primitive data, such as the raster images, could be in one of several representations to promote efficient storage. Multiple, complex representations of structure and data make searching difficult.

We do not intend our list of differences to be exhaustive, and without more experience, we would hesitate to try to quantify each of them. However, we feel that a design and implementation of a multimedia database must address these concerns.

IV. Use of ODA for Representing Multimedia Information

Given that we cannot use simple tables or networks to represent multimedia information, we need a richer representation. Although we could define a special purpose representation, it would not have wide support beyond our own effort. Therefore, we need to evaluate how some other representation might be used. Since we have already begun work using ODA as an interchange medium, it seems natural to evaluate ODA as a database representation as well. In this section, we describe enough of ODA to evaluate its applicability for use in a multimedia database system.

The ODA standard is quite large, and many of the details are orthogonal to any decision about using it as a database representation. However, there are several aspects of ODA that should be considered. First, ODA separates the logical structure of a document from its layout structure. Second, document architecture is separated from the content architectures of media. Third, one can define a generic structure that can constrain logical and layout structures. Fourth, ODA uses a pairs of names and values, called attributes, for labelling pieces of a document with information. Some of these attributes affect the use of ODA as a database representation. Fifth, attributes can be collected and shared through the use of styles. Finally, ODA defines an external (file) representation for documents. Each of these features is briefly described below.

The logical structure of a document within ODA consists of composite logical objects and basic logical objects that form a tree, rooted at a composite object termed the "document logical root" and with basic objects at the leaves. For example, a composite object might be a chapter or a section, while a basic object would refer to the text in a paragraph. The figure below shows part of the logical structure for a document.

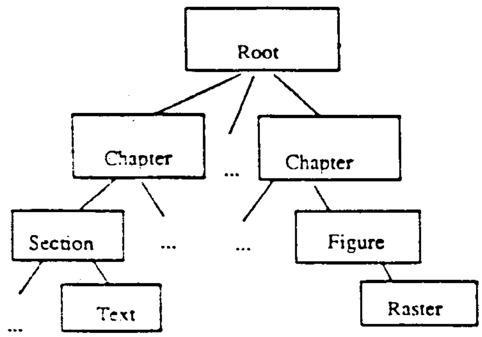


Figure 2: Logical Structure of an ODA Document (partial)

ODA does not define any particular kinds of composite or basic objects. It defines only a tree structure with internal nodes and leaf nodes. The labels of "Chapter" and "Section" are for illustration only and not part of ODA.

The layout structure of an ODA document consists of pages onto which the logical objects have been formatted. Adjacent parts of the logical structure may be formatted in vastly different parts of the layout stucture. For example, the logical structure can specify that a certain piece of text should be formatted in the table of content as well as appearing in the title of a chapter.

The separation of content architecture from document architecture allows for different kinds of information to be present for different media. For example, a description that a content has a certain pixel density only has meaning for a raster content, not for text. Similarly, the specification that paragraph indentation should be 1 cm only has meaning when formatting text. Through this separation, ODA allows a document structure to be defined independently of the kind of content attached to the leaves of either logical or layout tree.

As mentioned before, ODA does not define any particular kinds of structures in the logical or layout besides internal and leaf nodes. However, ODA does permit the definition of generic structures that can be used to constrain the way that documents are created and edited. For example, one can create a generic book which contains chapters, sections, and an index. By requiring the document production system to use the generic description of a book, one can ensure that newly created documents meet the specifications of the generic structure.

The information that describes objects in ODA is contained in attributes. For example, the selection of margin settings can be described with values for attributes. It is frequently the case that one wishes to collect together some attributes that have a related meaning. For example, one may want to describe the formatting of a quotation as changing to an italicized font, increasing the margins on both sides and centering the line. Although one could use the appropriate attributes everywhere a quotation is desired, ODA provides the ability to collect together these attributes into a style that can be referenced by all quotations in a document. In addition to saving space, the use of a style provides additional structure that can be used by database operations.

The most common external representation of ODA is called ODIF. ODIF is used to store an ODA document in a file or to sent it over a communications medium. ODIF is a context-sensitive, binary encoding of ODA structures based on the ASN.1 standard [ISO 8824]. It is difficult to produce, parse and manipulate. The other representation, called ODL, is not used anywhere to the authors' knowledge.

V. Use of ODA for Database Operations

With the brief description of ODA above, we can begin to evaluate it for use in a multimedia database by examining each of the factors listed in section III.

The first issue was one of sufficient number of primitive data types. Because ODA separates content architecture from document architecture, it permits a large number of media. A current limitation in the standard is that only three media types are defined: character, raster graphics and geometric graphics. There is work underway to expand the defined types to include equations, tables and additional media types. Although not a generous domain to work with, ODA provides more than simple text. As we will see when discussing expansion of primitive types, ODA can probably circumvent any problem with minimal primitive types.

The second criterion was that multiple presentations of data were needed. Unfortunately, ODA does not support multiple presentations of data. At best, ODA can be said to support a "logical" and a "layout" view of the same data, but the facility is content independent and not extensible.

There is a proposal to allow for alternative presentations of data that might satisfy this requirement. The current attributes for alternative presentations are quite primitive. In essense, they allow the substitution of a text string for a content. The motivation was that a comment, such as "figure showing grain sales", could be used if the content could not be processed. The two proposals for change are more general, but differ in where the alternatives may exist. One alternative is that an entire content portion may be exchanged for another. This would permit, for example, a raster to be substituted for a geometric graphic, or teletext to be substituted for a multifont text. The other proposal generalizes the first: it permits entire subtrees of logical and layout structure to be substituted for each other. Although both proposals support multiple presentations in a document, neither one addresses how more than one presentation could be examined (formatted, imaged) at the same time.

Obviously, direct support for multiple presentations of data would be useful in ODA. The group developing the table content architecture is also developing a collection of changes to support connections among various contents. The changes are intended to support exactly the relationships being discussed here. However, the current proposal is modest in comparison to state-of-the-art multimedia systems and is quite a ways from being adopted.

A third requirement for multiple descriptions of data has some support in ODA, but not much. One possible approach would be to use the same mechanism as for multiple representations. This approach has the same problem: only one description would be available at a time outside of the database application. A second approach would be to infer the necessary relationships between parts of a document through the use "indivisibility," "same layout object" and "binding" attributes. These attributes are used in concert to attach titles to figure, group related paragraphs, and generate page numbers and table of contents. Unfortunately, ODA defines only mechanism and not any use of the mechanism. Thus, one would need to establish a set of conventions that would list how certain bindings would be used. Then a database system could exploit this information.

A fourth concern was the need for expanding the primitive data types. The ODA standard has a provision for private content architectures, that is, for content architectures that are not defined in the standard. A private content architecture is one that meets the requirements of a content architecture, that is, defines coding attributes, presentation attributes, ODIF representation and formatting rules, but is not already in the standard. One may expand the primitive contents in this way while staying within the ODA standard. With this approach, one loses the ability to exchange private contents with systems that cannot process the information in the private contents, but otherwise retains compatibility with other contents. Further, the modularity of private content architectures increases the opportunities for adding any private contents to the ODA standard at a later date. We have done some preliminary investigation with private content architectures but it is too early to draw any conclusions.

The fifth problem was access to a large amount of structure in a document. Because ODA provides several structures describing the data, it is suited for representing multimedia data. Most of the structures are hierarchical; the support for arbitrary graphs is provided only for generic structures. However, the representation has the same problems and advantages as hierarchical databases: if you know where to find something, following links is fast, but if you do not know, doing a graph walk to search for it can be expensive.

The sixth concern was one of structure replication. Probably no single document will contain much repetition of structure. However, it is likely that structure among multimedia documents may be repeated. Such repetition can be captured in a generic ODA structure, which might be preprocessed for efficient searching of database requests. Another provision of the standard, Document Application Profiles, also can limit the kinds of structures that will be used in a document. It may be the case that a restrictive document application profile will be necessary to efficiently implement searching of an ODA document.

The final concern was one representation of data. The ODIF representation for documents is nearly useless for database applications. The encoding and low-level representation of document information are difficult to parse. One feasible approach is used by the MULTOS system: maintain a separate indexing structure for searches. Documents are preprocessed before being entered into the database. Searching for information is performed on non-ODA representations of the document while retrieval of the document is done using ODIF. We know of at least one other project that is pursuing the same strategy. Although the approach exploits many features of ODA, it has the severe problem that little sharing can be done among database systems except at the most primitive level of complete document exchange. There is no way to exchange indexing or preprocessing information, since it is unique to each system. What is clearly needed is some simple external representation of ODA that database systems can use. Unfortunately, there does not seem to be any such standard being developed.

VI. Conclusions

We believe that ODA provides some fundamental support for the use of media besides simple text, but ODA's structural features may be too general to allow for efficient implementation of search in a multimedia database system. It might be possible to exploit the generic structure of documents to speed the search for materials, but such a hypothesis needs to be tested empirically. Further, there is a clear need for an alternative document representation besides ODIF for file storage.

Because ODA provides a common way to interchange multimedia materials between systems, it is likely that it could be used as a communications medium between a multimedia database system and agents using the system.

VII. References

[Bertino]

Elisa Bertino, Fausto Rabitti, Antonio Converti, Angelo Giuntini, Pasquale Savino and Giancarlo Luxardo, "Text Retrieval Techniques for Office Documents," *Olivetti Res. and Tech. Review*, No. 6, 1986, p. 37-51.

[Borenstein]

Nathaniel Borenstein, Craig Everhart, Jonathan Rosenberg and Adam Stoller, "A Multi-media Message System for Andrew," *Proceedings of the USENIX Winter Conference*, February, 1988, pg. 37-42.

[Christodoulakis]

S. Christodoulakis, M. Theodoridou, F. Ho, M. Papa and A. Pathria, "Multimedia Document Presentation, Information Extraction, and Document Formation in MINOS: A Model and a System," *ACM Transactions on Office Information Systems*, Vol. 4, No. 4, October, 1986, p. 345-383.

[ISO 9541]

International Organization for Standardization, ISO 8613, *Information processing — Text and Office Systems — Office Document Architecture (ODA) and Interchange Format*, 1988.

[ISO 8824]

International Organization for Standardization, ISO 8824, *Information processing — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1)*, 1986.

[Palay]

Andrew J. Palay, Wilfred J. Hansen, Mark Sherman, Maria G. Wadlow, Thomas P. Neuendorffer, Zalman Stern, Miles Bader and Thom Peters, "The Andrew Toolkit -- An Overview," *Proceedings of the USENIX Winter Conference*, February, 1988, pg. 9-21.

[Reynolds]

J. Reynolds, J. Postel, A. Katz, G. Finn and A. DeShon, "The ARPA Experimental Multimedia Mail System," *IEEE Computer*, Vol. 18, No. 10, October 1985, p. 82-89.

[Rosenberg]

Jonathan Rosenberg, Mark S. Sherman, Ann Marks, Frank Giuffrida, "Translating Among Processable Multi-media Document Formats Using ODA," *Proceedings of the ACM Conference on Document Processing Systems*, December, 1988, p. 61-70.

[Thomas]

R. Thomas, H. Forsdick, T. Crowley, R. Schaat, R. Tomlinson, V. Travers and G. Robertson, "Diamond: A Multimedia Message System Built on a Distributed Architecture," *IEEE Computer*, Vol. 18, No. 12, December, 1985, p. 65-78.