

Exchanging Multimedia Documents: The Office Document Architecture (ISO 8613)

© 1989, Mark Sherman, all rights reserved

Although the Macintosh allows users to cut and paste between applications, the results are limited to either PICT, text, or private formats among the applications. As the number of multimedia systems grows, it becomes more difficult to exchange files and information. As one considers moving multimedia documents between Macintosh applications, *Presentation Manager* applications, and various workstation systems, the problems increase.

One solution is to use a standard intermediate representation. A recent ISO standard, the Office Document Architecture (ODA), provides a vendor-independent way to represent multimedia documents, facilitating their interchange. This article describes ODA and our experiences with it.

ODA provides a way to represent a structured, multimedia document. A structured document is one that is organized around either a logical partitioning, such as chapters, sections, and figures; or a layout partitioning, such as pages, header areas, and footnote areas.

Some Definitions

The logical structure of a document in ODA is represented by a specific logical structure, which is a tree. The internal nodes of the tree are called composite logical objects, and correspond to high level aggregations, such as a chapter or a titled figure. The leaves of the tree are called basic logical objects and contain content: multi-font text, raster images, or geometric graphics.

The layout structure of a document in ODA is represented by the specific layout structure, which is also a tree. The internal nodes of the tree can be page sets (collections of pages), pages, and frames. These features can be used

to implement various formatting strategies, such as collecting pages into an index, specifying multiple column layout, or positioning page headers. The leaves of the tree are called blocks and are used for positioning content.

Detailed information about a document is placed into attributes that are attached to the document's nodes (objects). For example, the attribute "indivisible" with the value "object type page" can be placed on a composite logical object and means that the entire subtree should be formatted on the same page.

The specific structures in ODA refer to a particular document. ODA also provides generic versions of logical and layout structures. The generic structures serve as prototypes or examples of documents. For example, a generic logical structure can define a "book" to be a table of contents, followed by any number of chapters, followed by an index. A particular book would have an actual table of contents, chapters containing the text of the book, and an index that refers to the text in the chapters. A generic structure for a page could describe the amount of room to be left for headers and footers, and the number of columns to be used. A specific layout structure for a page resembles a page-description language: the actual header, footer, and column texts would be formatted at particular places, ready to be imaged.

The formatting process in ODA takes a specific logical structure, an optional generic logical structure and a generic layout structure, to produce a specific layout structure. The document that is given to the formatting process is called a processable document, which means that it can be edited. The document that is produced is called a formatted document. A document may be both formatted and processable (which means that all four structures are present).

External Representation

ODA uses a binary encoding of a document for storage or transmission over a network. This encoding, called ODIF, is a compressed, context-sensitive, low-level representation of document information. In practice, it seems a difficult representation to work with.

Key Features of ODA

There are three key features of ODA: separation of specific structures from generic structures, separation of logical structures from layout structures, and separation of document architecture from content.

Generic structures provide a way to define a uniform collection of documents for an organization. For example, a company may define a standard interoffice memo, monthly report, and request-for-proposal document structure. An editor that understands the generic structure can ensure that all necessary parts of the document are provided and that it has a uniform appearance.

The separation of logical structures from layout structures allows one to format the same logical document for several uses. For example, one can use one layout structure for formatting a document for a computer terminal, another structure for hard copy.

The separation of document architecture from content allows one to extend ODA to handle new media types. Because the actual contents of a document are limited to the leaves of various structures, one can replace those leaves with new kinds of content while maintaining the rest of the structure of the document. ODA partitions attributes by use, and only presentation attributes apply to content. Each type of content has its own set of presentation attributes. Thus one can use ODA's provision for adding new kinds of content, such as equations and tables, by defining the representation of the information for that content (e.g., TeX for equations and SYLK for tables) and the presentation attributes that apply to that content. Plans are already underway to include spreadsheet data, video, audio, and other media into the standard.

Restricting ODA

ODA is too general for easy use; it is more of a framework to be applied. For example, the standard specifies that a document may consist of "composite logical objects." That these objects correspond to chapters or subsections is only a supposition in this article.

One way to limit the generality of ODA is through the use of document application profiles (DAP). A DAP restricts the ways that objects can be used. For example, one can define a DAP that specifies two superclasses: chapter and section. A document that conforms to that DAP would require that all of its composite logical objects be either chapters or sections.

To promote the use of ODA, standards bodies are working on a collection of DAPs for common use. The most common series is called the Q series, initially done by the Standards Promotion Application Group's (SPAG) ODA Expert Group and recently transferred to the European Workshop for Open Systems (EWOS) ODA Expert Group. Three of the DAPs, Q111, Q112, and Q113, are being used in Europe. Q111 roughly corresponds to teletext; Q112 provides features for writing business letters; Q113 defines structures for larger documents, such as numbered sections, headers, footers, and floating figures. Within the United States, NIST (formerly NBS) is developing a DAP called the ODA Implementors' Agreement, which is intended to be the same as Q113. A group of Pacific Rim countries are also developing a DAP that should match Q113.

EXPRES Project

Part of the National Science Foundation-funded EXPRES project (at Carnegie Mellon University and the University of Michigan) investigated interchange of multimedia documents among diverse systems. Along with our collaborators McDonnell-Douglas Corporation and NIST, we have built ODA support tools (called the ODA Tool Kit) and several translators among ODA, Diamond, *Interleaf*, *Andrew*, and *troff* formats. These were demonstrated at the ACM Document Processing Systems Conference in December 1988 and at the ODA

Workshop held at IBM's Almaden Research Laboratory in June, 1989.

ODA vs. SGML

Many view ODA and Structural General Markup Language (SGML) as alternative ways to represent documents. In fact, the two representations are incomparable. SGML provides a way to mark up a document. Thus, one could say that a chapter starts at some place in the text and continues until another place in the text. Similarly, one could mark where sections start and end. Each marked-up piece of text is associated with a tag. Just as ODA does not define any meaning for a composite logical object, SGML does not provide definitions for any specific tags. A document represented in SGML with an unspecified tag set is not processable except in the most abstract way. By contrast, ODA has a complete set of semantics of how a document should be formatted without need of a DAP.

The closest correspondence to ODA's DAP is a defined tag set, such as the one defined by the American Association of Publishers (AAP). This tag set is geared towards the transfer of manuscripts from authors to publishers. Thus, it can be used to denote chapter titles or section headings. However, there is no way to describe how a chapter should be formatted. In fact, the publishers do not want this information to be present since different publishers use different styles for their books. In SGML, there is no way to convey the formatting information. With ODA, one can choose to include it or omit it. Further, SGML does not define any media representations besides text.

Because SGML is so general, there does exist an encoding of ODA in SGML (called ODL).

The tag set used for ODL marks up a stream of characters into composite logical objects, tags where attributes begin and end, and indicates where generic structures begin and end. An ODA document encoded into SGML this way would be useless to a system expecting to find an SGML document representing the AAP set.

Caveats

This article is intended to provide the most useful information for someone who knows little or nothing about ODA. ODA is a relatively large standard. The main text of ISO 8613 is over 600 pages. Each content architecture refers to other standards for defining representations of the content information and formatting algorithms (character code and font standards for text, fax standards for rasters, and Computer Graphics Metafile—CGM—for geometric graphics). The external representation of ODA as ODIF relies on the ASN.1 standards.

Not all ODA structures, features, or their interactions have been discussed. The discussion of the structures has been simplified and in some cases, common computer terminology (tree, internal node, leaf) has been substituted for ODA's terminology.

If you want more information about ODA, you can get the standard from ANSI or get a copy of the book: *Multi-media Document Interchange: ODA and the EXPRES Project* by Rosenberg, Marks, and Sherman. The software developed at CMU as part of EXPRES—the Andrew multimedia system, the ODA tools and our ODA translators—is available on X tape from the X Consortium at MIT (the tools developed as part of the project can run on the Macintosh under MPW). You can contact me at the following address for more information:

Information Technology
Center

Carnegie Mellon University
4910 Forbes Avenue
Pittsburgh, PA 15213

412-268-6727

fax 412-268-6787

mss+@andrew.cmu.edu

