WSRC-IM--90-83-5

DE92 011375

A REVIEW OF COMPOUND DOCUMENT ARCHITECTURES (U)

by

B. J. Hudson

Westinghouse Savannah River Company
Savannah River Laboratory
Aiken, South Carolina 29808

Instruction Manual

DISCLAIMER

This manual was prepared in connection with work done under Contract No. DE-AC09-89SR18035 with the U.S. Department of Energy.

Reviews of Computing Technology:

A Review of Compound Document Architectures (U)

WSRC-IM-90-83-5

October 1 1991

Westinghouse Savannah River Company
P. O. Box 616
Aiken, SC 29802

Reviews of Computing Technology:

A Review of Compound Document Architectures (U)

by B. J. Hudson

WSRC-IM-90-83-5

October 1, 1991

Authorized Derivative Classifier

G. F. Hayhoe

Westinghouse Savannah River Company
P. O. Box 616
Aiken, SC 29802

Table of Contents

1 Introduction

1.1 Abstract

This review of computing technology will define, describe, and give examples of various approaches to document management through the use of compound document architectures. Experts agree that only 10% of business information exists in machine readable form, but much of what is stored is not in useful form. As a result, the average business document is copied over a dozen times during its life and duplicate copies are stored in numerous locations. The goal of compound document architectures is to provide an information support environment where rapid access to the correct information in the proper format is simplified. A compound document architecture provides structure to seemingly unstructured electronic documents, and standardizes the methods for interchange and access of entire or partial documents by authors and users.

1.2 Summary

Compound document architectures are not document processing software packages, but a methodology in which documents are easily interchanged, revised, and used in applications in addition to document processing. Information from documents in a compound document architecture can be automatically accessed by databases and document search software, and can be easily interchanged among document processing software. Multiple standards for compound document architectures exist, and are distinguished by orientation around unstructured documents such as mail, short reports, or correspondence where the physical layout is important (Office Document Architecture or ODA), or structured documents such as manuscripts or manuals where access to or interchange of content is important (Standard Generalized Markup Language or SGML). Deciding which architecture to use for a particular document type depends on how the information is used, both immediately and eventually, and whether the document must have the same appearance as the original.

Achieving the benefits of compound document architectures will require recognition and resolution of numerous issues, both administrative and technical:

- Document processing software packages need the ability to store and manipulate non-textual information.
- Data file format and interchange standards for compound documents must mature.
- Businesses must analyze documents to determine structures that enhance further electronic use.
- Authors and editors must recognize the separation of document structure, content, layout, and format.

2 General Discussion

2.1 Compound Documents in Today's Office

While the majority of document creation and editing involves only textual information, many new packages offer the capability of including graphic,illustrations, photographs, and other types of information in a document. Consider the creation, storage, transmission, and retrieval of a monthly report that contains text, a spreadsheet of expenditures, and a scanned photo of a new employee, and their presentation on a page with the new departmental logo. In today's environment, the data for this report would be generated from separate applications, and merged together with a document processor or desktop publications package, geared towards the creation of a formatted document on paper. Although the resulting file has unified the multiple forms of the information, it is very specific about layout, formatting, and fonts, and does not consider the eventual electronic use of the various components within it. The spreadsheet information has lost its identity with the creating application, and the text may have been copied from the word processor. Transmission of this file to another user would require that the recipient have the same layout software, and revision of non-text components might require extraction of the data back to the original application.

Since there is no recognized standard for document interchange in today's computing environments, intermediate exchange formats are necessary. The minimum expectation for document interchange is that all the information arrive without loss. Despite the reverse engineering of word processing native file formats and cooperation between product vendors, full fidelity formatted interchange is still unreliable. It is not always the fault of the interchange filter. Some recipient formats do not have the capability to accept particular features of the native format, and seemingly trivial issues such as font definitions can cause a document to paginate differently and appear totally different than the original. The use of filters between formats (for example, Word to WordPerfect) is cumbersome, and currently does not address multiple data types within the same document. Conversion also creates two copies of the document in different formats, thus causing debate as to which is the master copy.

2.2 Compound Document Architecture Concepts

A compound document processor integrates text, tables, graphics, data, images, and digital encoding of voice and video into a logical entity (a compound document), while maintaining each data type as an individual object. The methods by which this information is created, stored, transmitted, formatted, and presented constitute an architecture commonly referred to as a compound document architecture (CDA). Although there are industry and OSI standards and products associated with CDAs, the approach that an enterprise chooses in the management of document information requires consideration of the creation methods, interchange requirements, and archival and retrieval needs for the information.

Document storage and interchange must address both the capability to deliver a document to the recipient exactly as the originator created it (formatted form), as well as the capability for the recipient to further process it (processable form). Paper is considered the typical formatted form, while word processing native files are an example of processable form.

A CDA provides the higher level of integration in which the elements of the composite document are revisable as if still in their respective native applications, and are presented to recipients in a formatted form. (1) The various components are unified and can be edited, processed, and transmitted as a single unit, while the different components retain information about their individual data types. Depending on the design of the document architecture, the different data, text, and graphic components of a document may or may not all be stored in a single file.

One type of CDA defines a single file format which accommodates all potential data types, while others allow the launching of specific applications for particular data types, with the unification accomplished by a supervisory application. In any case, the contents of a compound document should be accessible to many applications across platforms, without conversion or corruption of the information. (2, 3)

If a compound document is treated as a collection of components with structural rules, an essential feature of compound document management is the separation of content from layout. A CDA treats a document as a logical collection of hierarchical objects such as chapters, sections, paragraphs, drawings, tables, and lists. The content resides within these contexts. "Document analysis" is used to define document types such as letters, reports, and memos used by organizations and to establish their required formats and layouts. For a CDA to be effective, the document standards must go beyond the definition of layouts to include the definition of the structuring rules for the objects—that is, the required and optional objects and allowable arrange-

ments and hierarchy of the components. The handling of document components as objects allows the creation of a conventional database structure for documents, opening the possibility of managing documents on an elemental basis rather than as long streams of unstructured data.

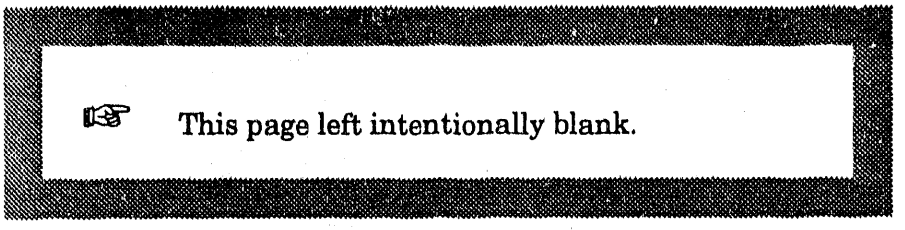
2.3 Benefits of a Compound Document Architecture

An important aspect of a CDA is that the information within documents is application-independent. A CDA defines a robust, documented, standards-based document format which can be processed by a wide variety of applications across platforms and vendors. The format becomes the least common denominator by which users exchange compound documents. Documents often outlive the word processing program that created them. With a CDA, the recipient of a document need not have the same word processing package to read or revise the document. If the concept of electronic records and the near-paperless office is to ever be achieved, the current need to store the applications software package that can edit the document along with the data must be overcome.

Document interchange extends beyond the exchange of single documents. Many business functions require the merging of sections from other documents, or the distribution of specific sections into text databases or other document applications. If the interchange model is extended to include delivery to other media (paper, CRT, image display terminals) and the full reproduction of graphical and geometric drawings, it is unlikely that interchange of documents can be accomplished without a standard, common storage and/or interchange format. A CDA must provide or define this format.

Compound documents facilitate the loading of complex, content-based retrieval systems. The internal tags used by a CDA for the linking and management of document components can be treated as field definitions for conventional database applications. Under a compound document strategy, documents become a database of self-indexed objects which can be processed, transmitted, and collected electronically into the form appropriate for the user's requirements. Consider the potential of searching documents for a particular keyword in an area as specific as the title of illustrations or an author's name in a footnote or references section. Also, the spreadsheet data from the past four monthly reports could be extracted from documents and collected into a new spreadsheet or spreadsheet component of a new document.

☞ This page left intentionally blank.



This page left intentionally blank.

An internal storage document exists physically as a single file, whereas a content pointer document is reconstructed from various components for the user to view. (5)

3.2 Office Document Architecture (ODA)

ODA/ODIF (ISO 8613) describes, the document interchange format that ensures receipt of a file which maintains the originators intent with regard to formatting and presentation, while maintaining the document's ability to undergo further revisions. The multi-part Office Document Architecture (ODA) standard was developed in Europe in the 1980s by ISO, International Telegraph and Telephone Consultative Committee (CCITT), and the European Computer Manufacturers Association (ECMA) for the processing and interchange of formatted documents typical in an office environment. The various parts of the standard define document structures, content structures, and the Office Document Interchange Format (ODIF). Different data types coexist within the document. A major emphasis in ODA/ODIF is to facilitate transmission of the file to another user, and to maintain document formatting and appearance just as the originator intended (4, 8). In 1991, seven major international computer vendors, including IBM and Digital, formed the ODA Consortium whose goal is to promote the ODA standard and products. GOSIP II (FIPS 146-1) requires that text processors adhere to the ODA standards. DOE has adopted the GOSIP standards, effective in 1992. Recent document application profiles (DAPs) define the level of functionality for document interchange for text, raster, vector, and graphical data. The three 1992 ISO DAPs under final review are FOD11—Simple document structure for character content only; FOD26—Enhanced document structure for character, raster, and geometric graphics; and FOD36—Extended document structure for character, raster, and geometric graphics. FOD26 will define the baseline for functions and features of personal computer document processing applications that can be interchanged.

3.3 Standard Generalized Markup Language (SGML)

SGML (ISO 8879), CALS, and FIPS 152 provide generic tagging of documents and management of pointers to external non-text files. An early initiative in compound document management was IBM's creation of a generic coding scheme for use in a law office information system and a scheme used by the Graphic Communication Association for book publishing and their submission to ANSI in the mid 1970's. After nearly a decade of work, ANSI and ISO issued the Standard Generalized Markup Language as ISO 8879:1986. It was immediately adopted by the American Association of Publishers (AAP) for book publishing and by the U.S. Department of Defense within its Computer

3 Standards Activity

Efforts on compound document architectures began over 10 years ago, and there are now two ISO standards for the representation and interchange of compound documents, as well as numerous industry trends. (4) The availability of compound document processors, architectures, and associated technologies has been driven by significant efforts on domestic and international fronts as well as industry trends. A discussion of the background of these major initiatives follows.

3.1 Approaches to Standards

Two approaches unify document components: internal storage and content pointers. If the full range of information types is unified within a document into a single file (internal storage) or collection of files (content pointers), the document can be edited, transmitted, or revised without regard to the data types.

An architecture that utilizes the internal storage of all data types has to provide applications support for interpreting and editing the binary representation of all potential forms of data within a document. This approach simplifies the file management issues, since the content and all related layout information (typically in a header) reside in a single file. It will, however, require extensions to the architecture for data types not conceived at the time the architecture was created.

A content pointer architecture stores much of the document content in a primary file and uses pointers to files and their associated applications or format standard for data types outside its defined scope. Upon display or revision of the document, the architecture issues a message to the associated application to display the referenced data from a standard format but within the physical boundaries of the display or electronic page. The potential also exists that if the referenced component is altered, the next time the document is accessed, the change will be evident to the user.

Aided Logistics Support (CALS) initiative for the production and transmittal of technical documentation and manuals. IBM still uses the early version of SGML internally for the majority of its publications. (6) In 1989, Federal Information Processing Standard (FIPS) 152 required that all Federal agencies specify ISO 8879 in text-processing procurements. DOE's Office of Scientific and Technical Information requires that electronic documents be transmitted as SGML by 1993. SGML utilizes generic tagging of text, and pointers to external data components (for example, graphics) coupled with accompanying document type definition (DTD) files which describe the layout and format of the final form document. A DTD may be written for any type of document, and different DTDs applied to a document will produce different output without the need to alter the SGML file whatsoever. (7)

3.4 Contrasting the Standards: Two Standards, Two Goals

ODA was created for office documents. While an ODA goal is to preserve original document formatting and layout, it is based on a flexible internal storaescheme which enriches the usefulness of a document beyond creation and revision. It was created to allow interchange of office documents which may vary widely in format and structure, and may even be free-form. It is not restrictive in the format and hierarchy of documents, but rather provides emphasis on the exchange of a document in a single file which completely describes and contains all elements and layout of the document. It assumes that the recipient will have access to a robust document processor or viewer with the same font sets and display capabilities as the originator. An advantage is that the exchange format is application independent so that it will outlive the current proprietary word processing approaches. Due to the lack of mandated structure in ODA documents, tagging that would provide capabilities for elemental searches and retrievals (for example, subheadings containing the word reactor) are not automatically defined nor easily attainable. Thus, effective retrieval of ODA documents will require content-based full text retrieval software in which a pointer to the compound document file is used for retrieval and browsing.

SGML was created for structured documents. SGML, by design, provides a structured tagging of every element of every document. A document must be assigned a Document Type Definition (DTD) before processing can begin. With each element of a document designated as a particular element type and format, the document is tagged and self-indexed as it is processed. Non-text parts of the document rely on an associated application which utilizes standard formats. For example, an illustration would be created in MacDraw, and that filename would be imbedded into the SGML document. The MacDraw information must be saved in a file in a standard format; otherwise, the application would need to be associated with the reference for the life of the document. Documents are easily transmitted between applications and platforms, but the DTD and supporting external referenced files must be trans-

mitted along with the document text file. (8) The recipient may apply the originator's or his own formatting rules via the DTD, and may create output in the preferred style.

The choice of ODA or SGML will depend on the application. There is no industry pressure to create a single compound document architecture standard. Office users prefer that the integrity of their presentation be maintained, but SGML users, such as publishers of large technical manuals, require the flexible tagging and separation of the various components of the documents (10). It is expected that some organizations will choose to use one architecture primarily, and will have little need for understanding or interchange into the other architecture. In large enterprises, however, it is likely that there will be workgroups (such as formal publications groups) that use SGML, and office workers who will use ODA. The interchange of documents between the two architectures becomes an issue in these cases.

3.5 SGML-ODA Interchange

Non object-linked ODA documents (those that reside in a single file) can be readily converted to SGML since they are completely self-contained and can only incorporate a predefined set of data types. A generic ODA-to-SGML document type definition (DTD) has been written but serves only to separate the various data types (text, image, graphics) into SGML elements and files. Hierarchical tagging of text is not fully accommodated, so the resulting document is not a robust SGML application.

SGML documents can reference any number of element types, many of which are not defined in the ODA model. Since SGML points to the elements of the document and their applications rather than containing them, consolidation of elements and subsequent revision of previous SGML elements within an ODA framework is unlikely. Thus, conversion of SGML documents to ODA will sometimes result in the loss of information. Examples of SGML elements for which there is no ODA counterpart are tables, processing instructions, and database references. (9)

4 Industry Trends

4.1 Personal Computer Word Processing

The de facto office infrastructure is the origin of a mass of documents that do not match the CDA models. The widespread use of personal computers for the input and processing of text documents and the current trend for document processors to allow the inclusion of graphical and other information have created a massive base for small scale document architectures. These compound documents typically live in separate applications on file servers or local disks, and are constructed into paper documents through layout programs or manual cut and paste. Almost all the applications follow the paper analogy, and do not propose to create document information in a form readily transportable to different applications, platforms, or non-page-oriented views.

Although the Macintosh computer is inherently graphical in nature, most of its word processors copy the graphics into non-revisable frames within the documents. Similarly, Microsoft Windows-based applications on the PS/2 workstation provide cut and paste capabilities of frozen views of spreadsheet or graphics between applications. Neither workstation has many products which address the compound document architecture requirement of logical structuring of objects within a document as addressable and revisable entities. Major vendors are developing extensions to existing word processing packages (planned for release in 1992) that will provide more complete compound document support

4.2 CDA at the Desktop

It has been reported that major word processing vendors, particularly those with major European markets, are producing options in their products which will allow users to save documents in ODA format. For example, when WordPerfect and Microsoft Word both have ODA options, there will be hope for full-fidelity document interchange between users. The features supported in interchange are described in FOD26. (10) Additionally, third parties such as

Keyword Technologies are producing ODA converters from a wide variety of word processing native file formats. (11) This after-the-fact approach to conversion typically does not produce reliable conversions, primarily due to the fact that the authors did not create the document with ODA output in mind, thus omitting the document analysis useful in creating flexible, "smart" documents. Initial releases of the ODA options will probably not support non-textual components.

DEC, Apple, and IBM are heavily committed to products associated with ODA due to their major segments in the international and European markets. Since they are large corporations, they also provide offerings in the SGML area. The Digital Compound Document Architecture Strategy is ODA-based, with proprietary extensions, and includes a suite of products including document processors and conversion software. Apple has recently announced the availability of the WOPODA toolkit which can be utilized by developers as an interface for non-ODA compliant document processors to produce ODA files for interchange. While IBM recently enhanced their Mixed Object Document Content Architecture (MO:DCA) to be ODA-compliant, there is not a range of products available that fully utilize the ODA capabilities. In most cases, the native form of documents is at least partially proprietary, but the endorsement of conversion to Office Document Interchange Format (ODIF) for interchange of documents is encouraging.

There is recent activity in the Microsoft Windows community to create the Compound Document Protocol Standard (CDPS) as an extension of Object Linking and Embedding (OLE). OLE resembles an object-pointer CDA. Users will be able to launch the application for non-text frames from within the document processor. The file edited by the non-text application will actually live outside the document, and changes to it will be reflected in the document (even if the changes were made outside the document processing environment). Windows does not manage the links, nor can it recreate a document if the filenames are changed or moved to other servers. It does not provide the capability to make a copy of the non-text file for the document's exclusive use if that requirement is desired. The CDPS will be an attempt to manage the various elements of a document and ownership and document layout issues, as well as describing the interchange formats between CDPS compliant applications. Most major Windows applications are expected to support OLE during 1992, and the CDPS will further stabilize by late 1992. (12)

5 Planning Compound Documents for Electronic Use

Documents usually require various representations for different users. Early document architectures recognized that the form which is finally presented (currently a format for a laser printer) is not appropriate as the format in which the editing and revisions occur. For example, PostScript printers create a bitmap of the page representation, but the revisable form of the document is a binary file of coded textual characters. Few document processors have layout capabilities for formats other than paper, for example presentation on computer terminal screens. Neither the revisable nor the output form of a document is appropriate for use in loading information retrieval systems which would allow searching of the content of documents and the retrieval of specific sections, illustrations, or graphics. To further complicate the issue, interchange formats are often required because the revisable form is incapable of being transmitted by conventional methods due to their internal binary codes. Thus, electronic document managers have been faced with the dilemma of maintaining numerous copies of documents in different formats for specific intermediate and end uses.

Document analysis can identify finer granularity for access to specific document content. A well written document contains an information hierarchy, often visualized in the table of contents. For example, there are chapters, sections, then sub-sections and paragraphs; each titled component would be reflected as an entry in the table of contents. The logical representation of a document can facilitate access to specific content and can imply an importance level and interrelationship to other document parts based on its location and presentation in the table of contents. Large documents also often have an index. This allows access to specific content based on concepts or specific words. Neither of these concepts is dependent on the visual appearance of the information itself.

If this table of contents and index analogy is carried into the document processing world, originators of documents should be less concerned with layout, and more concerned with content and structure. A new concept associated with compound documents is "document analysis." It is a methodology for identifying document hierarchy, and conventions for document cross-refer-

ences, section naming and numbering, and document presentation and formats. Many businesses are identifying standard document types and defining layout standards as well as required logical hierarchy within documents. The various compound document architectures have guidelines for describing and creating electronic document type standards. These descriptions act as interpreters of the input information and provide visual attribute information to translators which format the document to appropriate target devices, whether paper, a database, a document processor, or a video screen. The content file is the single stored file for the document, and the formatter provides the appropriate view, depending on the intended use of the information. (6) With proper planning, a single file can provide views of documents for all uses and can be accessed from independent applications.

6 Elemental Tagging Concepts

Within a CDA, the information hierarchy and document logical structure is accomplished by internal tagging and marking; these tags do not specify visual attribute information. Visualization and layout are handled upon request by associated applications, one of which is the document display processing package. The creation tool guides the author through the appropriate levels of hierarchy, allowable element types, and logical structure of the document type being edited. It is user friendly and displays a document as it will appear in a selected final form, but that final form need not be limited to paper. The final layout and document structure are governed by a pre-determined document definition or enhanced style sheet. For example, the authoring tool will appear to mark the title as bold, 18 point, but it will actually know only that it is a title, and the current defined view for a title requires it to be displayed in that fashion. In another context, such as a database load, the font for the title is irrelevant, but the fact that it is a title is extremely important as a database field (13).

6.1 Specific Tagging

The following section is adapted from the annotated ISO 8879 introductory materials (6). A similar internal tagging is used in ODA. The method by which document processors internally represent document layout and structure can affect the subsequent flexibility of the use of the document content. For example, the introduction to a manuscript might look like Figure 1 when printed. It has a bold, larger, centered title; bold, numbered section titles; paragraphs separated by blank lines; an item underlined for emphasis; and a list of items.

Scan Test Results and Recommendations

1.0 Overview

There are many parameters to be considered when setting up a scanning procedure for documents. This study examined the impact of scanning density, contrast control, and compression methods on file sizes and retrieval times for the Image System.

There were 3 assumptions:

a. Workstations have 200 dpi screens

b. The printer is 300 dpi

c. The miminum acceptable retrieval time for a document page shall not exceed 4 seconds

2.0 Summary of Conclusions and Recommendations

Assuming a decision can be made as to the primary purpose of the scanning effort (whether to produce quick-retrieval images at screens or good quality hardcopy), there are clear indications regarding scanner settings.

Figure 1: Printed document output

Tagged specifically, it might be internally represented as illustrated in Figure 2 (the tags are ***bold italics***):

```
.bc HV14 Scan Test Results and Recommendations
.lmar 5 .rmar 50
.bl HV 10 1.0 Overview

.pl TI 10 There are many parameters to be considered
when setting up a scanning procedure for documents.
This study examined the impact of scanning density,
contrast control, and compression methods on file
sizes and retrieval times for the .uImage System.u.

There were 3 assumptions:
.lmar 7 .rmar 40
.hng 10
          a. Workstations have 200 dpi screens
          b. The printer is 300 dpi
          c. The miminum acceptable retrieval time for
a document page shall not exceed 4 seconds

.lmar 5 .rmar 50
.bl HV 10 2.0 Summary of Conclusions and Recommenda-
tions

.pl TI 10 Assuming a decision can be made as to the
primary purpose of the scanning effort (whether to
produce quick-retrieval images at screens or good
quality hardcopy), there are clear indications
regarding scanner settings.
```

Figure 2: Specific document tagging

Figure 2 describes the attributes of a section or a paragraph for each occurrence, including its font (***.TI 10***) and left justification in plain typeface (***.pl***), and to explicitly number the sections and the list items. Also, the margins (***.lmar 7*** and ***.rmar 40***) and the hanging indent (***.hng 10***) are based on the predicted size of the font. The underline attribute is specifically coded (***.u***). Blank lines were manually inserted between the paragraphs. When the paragraph in section 2 was input, it required a restoration of the margin and font parameters of the previous paragraphs through explicit definition (***.lmar 5*** and ***.rmar 50***).

6.2 Generic Tagging

Generic tagging defines and names permissible document elements and their attributes, and is preferred for internal document representation in CDAs. A document is described in terms of these definitions. This approach avoids the repetition of the specific descriptions, and eases the changing of an element type throughout a document. Tagged generically, the internal representation of the sample might be as shown in Figure 3.

```
<doc>
<ti>Scan Test Results and Recommendations</ti>
<sec>Overview</sec>
<par>There are many parameters to be considered when
setting up a scanning procedure for documents. This
study examined the impact of scanning density, con-
trast control, and compression methods on file sizes
and retrieval times for the <emp>Image Sys-
tem</emp>.</par>
<par>There were 3 assumptions:</par>
   <lis>Workstations have 200 dpi screens</lis>
   <lis>The printer is 300 dpi</lis>
   <lis>The minimum retrieval time for a document page
shall not exceed 4 seconds</lis>
<sec>Summary of Conclusions and Recommendations</sec>
<par>Assuming a decision can be made as to the
primary purpose of the scanning effort (whether to
produce quick-retrieval images at screens or good
quality hardcopy), there are clear indications regard-
ing scanner settings.</par>
</doc>
```

Figure 3: Generic document tagging

The tagged version in Figure 3 generalizes the datafile, and has an associated definition file (Figure 4) which is be used to describe the layout attrbutes of each defined element. There are no explicit font or ruler definitions; they are provided for each type in the definition file. The numbered lists and sections will be numbered or lettered as sequenced by the formatter in preparation for presentation. Generic tagged files allow changes to apply to the entire document without editing the content file. Associating the sample DTD in Figure 4 with the generic tagging in Figure 3, see that the blank lines between sec-

tions and paragraphs will be inserted by the formatter depending on the ara and element specifications in the description file which is associated with this document. If the document were to be formatted for dual column layout, only the definition need be changed; similarly, if the chosen form of emphasis changes to italics or perhaps a color for a video display, just the description of the attribute in the document definition would be altered and would apply to all instances in the document. This source file in Figure 3 contains only the information in a logical format useful to a wide range of applications, while the DTD describes the layout and formatting. Further, the entire file is ASCII, which is transportable to any platform.

```
<!doc> <% pagesize "8.5 11" % columns "1">

<!ti> <% font "Helvetica 14" % margins "5 50" % trailline "2">

<!sec> <% font "Helvetica 10" % margins "5 50" % numbered "0.0">

<!para> <% font "Times 10" % margins "5 50" % headline "1">

<!emp> <% font "Helvetica 10 underline">

<!lis> <% font "Helvetica 10" % margins "7 40 10" % numbered "a.">
```

Figure 4: Sample document type definition (DTD)

☞ This page left intentionally blank.

7 Summary and Conclusions

A goal of compound document architectures is to provide an information support environment where rapid access to the correct information in the proper format is simplified. Conducting document analyses and selecting a compound document architecture will enhance document creation, revision, and storage methods so that information can be retrieved in the form needed through intelligent retrieval methods.

Both SGML and ODA provide guidance for standards associated with compound documents, but are designed with different intents. There is enough activity in these standards and in emerging desktop document processing application software to warrant pilot projects. Future procurements should consider the applications and vendors which have indicated a commitment to the standards.

The next generation of desktop document processors is likely to provide a "Save as ODA" option or ODIF conversion routines. They will create true compound documents that incorporate non-text information using ODA techniques rather than freeze-frame or manual cut and paste. Enhancements to software packages, coupled with the creation of guidelines for routine office correspondence such as reports, will facilitate document interchange and "roll up" into other documents.

SGML is designed for documents such as those produced by formal publications groups, particularly long documents or those whose content might be later presented in another document with a different style. SGML will enhance the ability to gain access directly to a section or topic of interest, or even automatically extract the portions of documents applicable to a particular interest area. SGML tags can be used as tags for hypertext, BASIS, or other full-text retrieval systems.

To address the issues associated with compound documents, users and planners should be prepared to:

- Limit document processing packages to those that will provide the capability of producing ODA-compatible PC and Macintosh formats.

- Begin analysis of documents to determine appropriate means for separating content from layout, and to assign hierarchies to the information.
- Include document analysis in document processing user training courses.
- Pilot SGML for a major class of structured documents for which intensive retrieval is likely.
- Begin to utilize the electronic form of documents as the first means of information access, rather than reliance on paper copy.

References

1. "DEC's CDA: More Than Just a Pretty Face," *Office Information Systems*, C-DEC-676.1, Gartner Group, Inc., January 15, 1990.

2. *VMS Compound Document Architecture Manual*, Digital Equipment Corporation, Maynard, MA., December 1988.

3. "Electronic Imaging," *Office Information Systems*, OIS: R-100-109, Gartner Group, Inc.

4. "ODA: What Is It? What Good Is It?," *Seybold Report on Publishing*, December 18, 1989, p. 3.

5. "NewWave and CDA—Complementary Architectures," *Office Information Systems*, P-014-715, Gartner Group, Inc., April 30, 1990.

6. Charles Goldfarb, *The SGML Handbook*, Oxford, England, Clarendon Press, 1990.

7. "New Federal Information Processing Standard: FIPS 152, K-014-602.1," *Office Information Systems*, Gartner Group, Inc., May 19, 1989.

8. "ODA: The Other Document Architecture," *Corporate Publishing Strategies*, T-018-123.1, Gartner Group, Inc., March 31, 1989.

9. Sharon Kemmerer, ed., *Collection of Technical Studies Completed for the CALS Program—FY 1987*, Vol. 1, NBSIR 88/3276, pp 1-16, 52-80, March 1988.

10. *Draft Working Implementation Agreements for Open Systems Interconnection Protocols*, Parts 16 and 17, Gaithersburg, MD, IEEE, September 9, 1991.

11. Discussions with Mike Howard of Gartner Group, Inc., August 1991.

12. Gina Smith, "New Linking Technology Lets Different Data Formats CoMingle (OLE)," *PC Computing*, April 1991, p. 44.

13. SGML Presentations by Thomas Tallant, Oak Ridge National Laboratory, July 1991.

END

DATE FILMED

5/14/92