ODA: A Study of Document Design

R. N. Jugele\(^1\) and Dr. V. N. Chavan\(^2\)

\(^1\)Department of Computer Science, Science College, Congress Nagar, Nagpur, Maharashtra

\(^2\)Head, Department of Computer Science, S. K. Porwal College, Kamptee, Dist: Nagpur. Maharashtra

Abstract: A document is considered as a textual record. A unit consisting of dynamic, flexible, nonlinear content, represented as a set of linked information items, stored in one or more physical media or networked sites, created and used by one or more individuals in the facilitation of some process or project. In ODA document architecture is defined as the set of rules for structuring interchanged document. The logical structure divides and subdivides the content of a document into increasingly smaller parts on the basis of the human perceptible meaning of the content e.g. into chapters, sections, paragraphs and figures. The layout structure divides the content into sets of pages, individual pages and areas within pages such as columns. Each structure represents a different but complementary view of the content of a document. Each structure is a hierarchy of objects, represented by sets of attributes that define the properties of the objects and their relationships.

Keywords: Architecture, Object, Style, logical, Layout, Generic, Content, ODA.

1. INTRODUCTION

Open Document Architecture (ODA), gives the way to represent and exchange structured, multimedia information to the growing need for heterogeneous systems to exchange documents containing multi-font text, graphics of various kinds and layout. It provides ways to go beyond text and graphics to expand the available media to accommodate audio, video and other content. The standards [3][10][1][9] provide the detail view. It represent a structured document that is organized around either a logical partitioning of a document such as chapters, sections and figures or a layout partitioning of a document such as pages, header and footnote areas. Multimedia document contains multi-font text, raster images and geometric graphics [11].

The logical structure is represented by the specific logical structure which is a tree and the internal nodes of the tree and it is called composite logical objects which correspond to aggregations such as a chapter or a titled figure. Leaves of the tree are called basic logical objects and contain content i.e. text, raster images or geometric graphics.

The layout structure is represented by the specific layout structure which is also a tree and the internal nodes of the tree can be page sets, pages and frames. These correspond to formatting strategies i.e. collecting pages into an index, specifying multiple column layout or specifying the position of page headers. The leaves of the tree are blocks which is used for positioning content.

Complete information about document is placed into attributes and attached to the documents objects. The specific structures refer to a particular document. The ODA provides generic versions of logical and layout structures which serve as prototype. For example a Book is a table of contents having any number of chapters and an index. A particular book have an actual table of contents, chapters containing text of book and an index refers to the text in the chapters. A generic layout structure for a page describe the amount of room to be left for headers, footers and number of columns used. A specific layout for a page have a page description language i.e. actual header, footer and column text to be formatted at a particular places.

ODA uses a binary encoding for storage or transmission over a network called ODIF/ODL, it is a compressed, context-sensitive and low-level representation of document information based on ASN.1.

ODA is the application-layer standards in the OSI model [7] which is used to represent multimedia messages within CCITT standards via the aligned T.410 series of recommendations[8]. It is one of the representations used for vendors to exchange graphical information for the CALS project of Defense to enable and accelerate the integration of digital technical information for weapon system acquisition, design, manufacture and support [12][13][5]. It species cut and paste format that can be used by multimedia applications running under the X Window System [2]. It is proposed as one of the representations for medical record information within the IEEE MEDIX standard [6] for hypermedia information in the Dexter Hypertext Proposal [4] and for document interchange [14]. It is adopted in a variety of other standards to meet set of needs.

ODA is a very large and general standard thus different user communities defines an appropriate subset of ODA which fulfills their needs, it is called Document Application Profiles (DAP). A cooperative effort involving industry, European government and academia investigated the implementation and application of ODA while it was being developed to ensure ODA feasibility. ODA is a mechanism which can be used to help integrate different sources of multimedia information.
2. ODA

To exchange, process and present documents in open distributed systems is the main objective of the ODA standard. It is based on data model which distinguishes between the content which comprises all information that is presented to the document user and the content portions consist of information of exactly one content type and the structure of a document. ODA considers three contents namely character text, raster images and geometric graphics. For every type a content architecture is standard, consist of a specification of the types elements, control functions applicable to them and encodings used. It also determines the formatting and imaging process for these elements.

Structure of a document consist meta-information about the document content useful for editing and formatting. Further document is divided into objects of a logical structure and a layout structure as shown in figure 1. The logical structure of a document results from the editing process and defines the document components meaningful to the user, dividing the document into objects. ODA does not define the objects that are part of a logical structure. The layout structure of a document results from the formatting process and groups the content portions of the document into presentation units, distinguishing between objects. ODA defines fixed set of objects which are part of the layout structure and identifies their potential relationships.

The logical and the layout structure are trees in which the objects constitute the nodes. The leaves of the trees are called basic objects and associated with content portions of the document and determined by attributes associated with each node.

Each attribute defines a certain type of tree construction:
- **SEQUENCE**: Arguments are sequentially ordered
- **AGGREGATE**: No particular ordering among the arguments
- **CHOICE**: One of the arguments is chosen

Each attribute prefixed by one of the construction factors:
- **OPTIONAL**: Argument may or may not be included in the structure
- **REPEAT**: Argument may be included in the structure one or more times
- **OPTIONAL REPEAT**: Argument may be included in the structure zero or more times

If there is no construction factor specified then the argument is included in the structure exactly once. Attributes can refer to the tree construction as well as to other properties and a typical attribute of a frame would be its spatial position within a page. Attributes can also be used to associate objects of different structures with each other. To determine attributes resulting from the formatting process then the user can specify formatting directives resulting in certain styles.

Layout styles specify rules or restrictions for mapping logical objects onto layout objects. It describe that a figure and its caption should appear on the same page.

Presentation styles are associated with basic logical or layout objects and define content specific formatting attributes such as the font to be used for displaying text by changing its style definitions. A document can be presented in a variety of ways but its content and logical structure remain unchanged.

ODA also permits the definition of generic structures which can be used to specify document classes. A document class comprises documents with identical structural characteristics. For example technical reports of a project can be considered as instances of the same document class. They all have a similar logical structure and shall be formatted in the same way. They may even share standard content portions such as the project logo or the list of sponsors. Similarities can also be seen within a document i.e. each paragraph should be handled in the same manner. This is expressed by grouping objects into object classes. Generic mechanisms save the author specification work, particularly in defining document styles. They also allow more efficient document encoding and exchange.

![Diagram](image.png)

**Figure 1**: Process of Formatting and Presentation

### 2.1. ODA Features

ODA is designed to facilitate transmission of compound documents between open systems. It focuses on blind interchange i.e. the originator need not know anything about the recipients system. An ODA document may easily be transferred from one word processor to another and this is called revisable or processable form. It also supports a final or formatted form i.e. the receiver cannot edit the document.

### 2.2. Organization of ODA

The ODA standard divides information into three main categories:
- **Logical information**: It is the structuring of the content in terms of hierarchy and order. e.g. a chapter may be seen as a sequence of sections which in turn may be a sequence of paragraphs.
- **Layout information**: It organizes the physical appearance of the content on a presentation medium. ODA defines a hierarchy of layout components called page sets, pages, frames and blocks.
- **Content**: It organizes portions of either text, images or graphics. These content portions are references from both the logical and layout structure and make it possible to either have a logical view or a layout view of the content. Different views of the document facilitates different applications. A printing application would only need the layout view in order to construct an image of the document while a word processor would use the logical view in the editing process as shown in figure 2.

ODA also has three more components:
Generic structures: It can be logical or layout. These are rules which define the class of a document. For example a class “article” may determine that the document must start with an abstract, followed by one or more numbered chapters each consisting of only one level of sections.

Styles : ODA has divided the style concept into -
- Layout styles: It is associated with the logical structure. It can specify that a heading and the following paragraph both should appear on the same page.
- Presentation styles: It is concerned with the layout and imaging aspects of the content and are specified for the lowest level logical and layout components. There are different sets of presentation styles for different content types. For character content one may specify parameters like line spacing or which fonts to use.

Document profile: It contains information about the document as a whole. It manages information (e.g. title, name of the author, keywords) and technical information like which structures are present and which coding standards are used for different content types.

3. DAP
ODA specifies a way to form subsets of the total set of features to implement different levels of user requirements, these subsets are called Document Application Profiles (DAP).

There are three steps of profile features:
- The first step provides for documents containing character content only. The document may have sequences of paragraphs which are laid out in a single column of text.
- The second step supports documents with both character, image, graphics content and can be structured into chapters, sections and paragraphs. The content may be laid out in multi-columns.
- The third step provide support for more sophisticated word processing.

4. ODA MODEL EXAMPLE
Document Structure:
Document is interchanged to contain processable information. In ODA this is represented using the logical structure, figure 3 shows the ODA structure. Here assume that there is a parent component i.e. a composite logical object and the children of this parent are also to be composite logical objects.

Figure 2: Processing Model of ODA

Logical Object definitions:

SECTION = {
$CLASS,
Superclass = COMPOSITE_LOGICAL,
Attributes = {
$DEFINE,
Content_Type = PARAGRAPH
}
}

CHAPTER = {
$CLASS,
Superclass = COMPOSITE_LOGICAL,
Attributes = {
$DEFINE,
Content_Type = SECTION | PARAGRAPH,
Chapter_Number = NUMBER,
Chapter_Title = STRING
}
}

Object Class and a Style:
The ODA structure shown in Figure 4 have three constituents:
- Basic logical object: It indicates that it is an instance of the basic logical object class by the object class attribute
- Basic logical object class: It has an associated presentation style as indicated by the presentation style attribute
- Presentation style: It has one attribute associated with it, the character content architecture attribute indentation and the indentation attribute are according to ODA semantics and DIL.

Figure 4: ODA Object Class and Style
Logical Object Class and Style:

```plaintext
LOGICAL = {
    $CLASS,
    Superclass = DIL_OBJECT,
    Attributes = {
        $DEFINE,
        Protection = BOOLEAN,
        Layout = ANY,
        Breakable = BOOLEAN
    }
}

COMPOSITE_LOGICAL = {
    $CLASS,
    Superclass = LOGICAL,
    Attributes = {
        $DEFINE,
        Logical_CONTENT = COMPOSITE_LOGICAL | BASIC_Logical
    }
}

LOGICAL_ROOT = {
    $CLASS,
    Superclass = COMPOSITE_LOGICAL
}

DOCUMENT = {
    $CLASS,
    Superclass = LOGICAL_ROOT,
    Attributes = {
        $DEFINE,
    }
}

BASIC_LOGICAL = {
    $CLASS,
    Superclass = LOGICAL,
    Attributes = {
        $DEFINE,
        Content_Type = ANY
    }
}

PARAGRAPH = {
    $CLASS,
    Superclass = BASIC_LOGICAL,
    Attributes = {
        $DEFINE,
        Content_Type = CHAR,
        Left_Margin = NUMBER,
        Right_Margin = NUMBER,
        Widow_Size = NUMBER,
        Orphan_Size = NUMBER,
        Tab_Stops = TABSTOP
    }
    style = {
        $DEFINE, Left_Margin = LM
    },
    para1 = {
        $PARAGRAPH, style,
        {
            $CHAR, "Paragraph One"
        }
    },
    para2 = {
        $PARAGRAPH, style,
        {
            $CHAR, "Paragraph Two"
        }
    }
}
```

5. ADVANTAGES

Following are reasons that ODA is a more suitable format for document distribution:

- A single ODA file can encapsulate a compound document, its distribution only requires a single file to be passed.
- The ODA file contains enough information to render the file on screen or paper in a pleasing and meaningful manner.
- ODA can be readily converted into a wide range of commonly used word processing formats. It is possible for a system which holds documents in ODA to deliver them to users in a format which they can view on their normal equipment they can edit these documents annotate them or extract parts into their own documents all within their normal document processing environment.
- The ODA format is reasonably compact. The format supports geometric graphics and bitmaps are compressed using the Group 4 fax algorithm an excellent lossless compression scheme or Group 3 fax algorithm or bitmap.
- The ODA format does not suffer the ASCII-related problems. Its file do not need altering when files are transferred between ASCII and EBCDIC based machines or between machines with different byte orders or between ASCII based machines with different line break characters.
- There are limits in the flexibility of changing font sizes in ODA.

6. DISADVANTAGES

The main reasons of ODA failure can be:

- The standard was too ambitious for the computer capabilities at the time.
- It was too complex to adopt and remained theoretical with no tool support.
- The arrival of more pragmatic standards shifted support away from ODA.
- The generalization and evolution of RTF from Microsoft.
- The standard grew out of control.

6.1. Technical Drawbacks of ODA

- The editing structure could not be preserved between many word processing applications.
- ODA did not provide the same set of primitive descriptions of data.
7. CONCLUSION

Document structures have led to many different systems for handling structured documents. Existing systems provide many of the advantages but are generally poor at representing complex or dynamic relationships between components of a document. It is designed to facilitate inter-operability between different document processing systems and document interchange occurs whenever one person sends a document to another. Users may prefer electronic document interchange for a variety of reasons. The essential benefit within the context of document interchange is to give the document vendor independence, because ODA standard is written to have a very broad scope. However no device can support every possible feature. The set of defined document application profiles (DAP) are arranged in levels of increasing functionality. Each DAP defines a list of supported features which any system at the same or higher level must be able to accept or interpret correctly.

REFERENCES


Books:

AUTHOR
R. N. Jugele received his B.Sc. degree from Nagpur University and M.Sc. degrees in Computer Science from Marathwada University, Aurangabad, Maharashtra, India in 1991 and 1993 respectively. Currently, he is working as a Associate Professor in Department of Computer Science, Science College, Congress Nagar, Nagpur, Maharashtra.

Dr. V. N. Chavan Head, Department of Computer Science, S. K. Porwal College, Kamptee, Dist : Nagpur, Maharashtra. He is research guide in Computer Science subject in various universities and have wide knowledge in research field since last 22 years. He is a member of various bodies in Computer Science subject.