Chapter 8

Documentation, Hypertext and MHEG

Documents

A document consists of a set of structural information that can be in different forms of media, and during presentation can be generated or recorded. A document is aimed at the perception of a human, and is accessible for computer processing.

A multimedia document is a document which is comprised of information coded in at least one continuous (time-dependent) medium and in one discrete (time-independent) medium. Integration of the different media is given through a close relation between information units. This is also called synchronization. A multimedia document is closely related to its environment of tools, data abstractions, basic concepts and document architecture.

Document Architecture

Exchanging documents entails exchanging the document content as well as the document structure. This requires that both documents have the same document architectures are the Standard Generalized Markup Language (SGML) and the Open Document Architecture (ODA). There are also proprietary document architectures, such as DEC's Document Content Architecture (DCA) and IBM's Mixed Object Document Content Architecture (MO:DCA).

Information architectures use their data abstractions and concepts. A document architecture describes the connections among the individual elements represented as models (e.g., presentation model, manipulation model). The elements in the document architecture and their relations are shown in Figure.

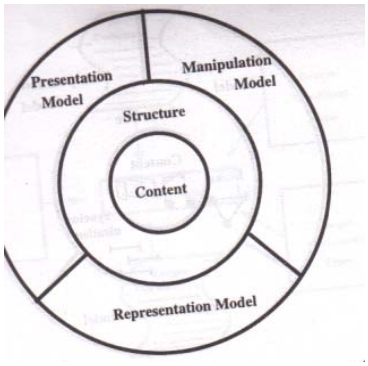

Figure: Document architecture and its elements.

The manipulation model describes all the operations allowed for creation, change and detection of multimedia information. The representation model defines: (1) the protocols for exchanging this information among different computers; and, (2) the formats for storing the data. It includes the relations between the individual information elements which need to be considered during presentation. It is important to mention that an architecture may not include all described properties, respectively models.

Manipulation of Multimedia Data

The user becomes most aware of multimedia documents through tools for manipulation of multimedia data, such as editors, desktop publishing programs and other text processing programs.

A document undergoes the process shown in Figure. The information included in a document belongs to a certain document type, e.g., a business letter or an internal memorandum. The same document can belong to other types which mainly influence the final representation. The transformation from the actual information to its final representation behaves according to rules specific to the document architecture.

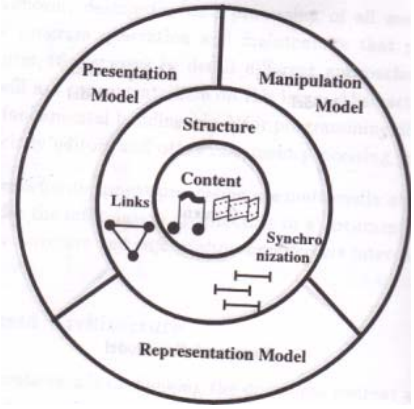

Figure: A multimedia document architecture and its constituent elements.

The processing cycles of a traditional document and an interactive multimedia presentation are analogous, as shown in figure.

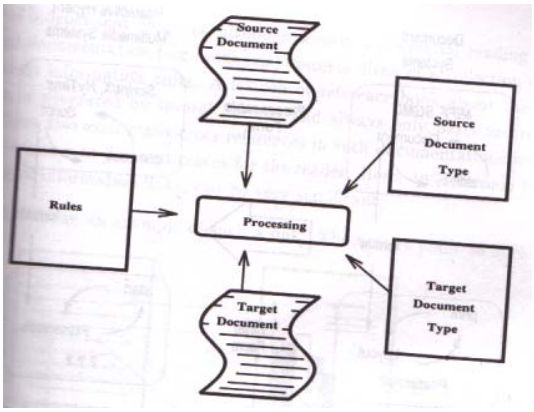

Figure: Processing of a document: from the information to the presentation

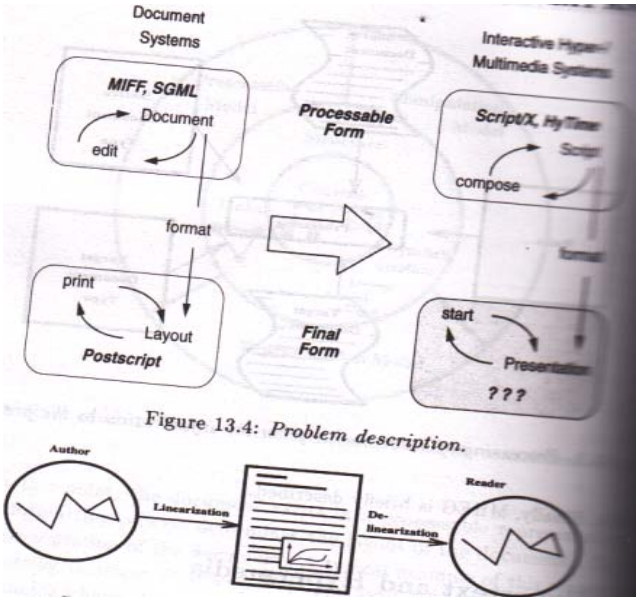

Figure: Information transmission

Currently, an author edits a document with a text editor. Thus, he or she uses the system's character set (e.g., ASCII) as the actual content of a document, as well as a hidden language available in most interactive editors for structural description (e.g., SGML).

Hypertext, Hypermedia and Multimedia

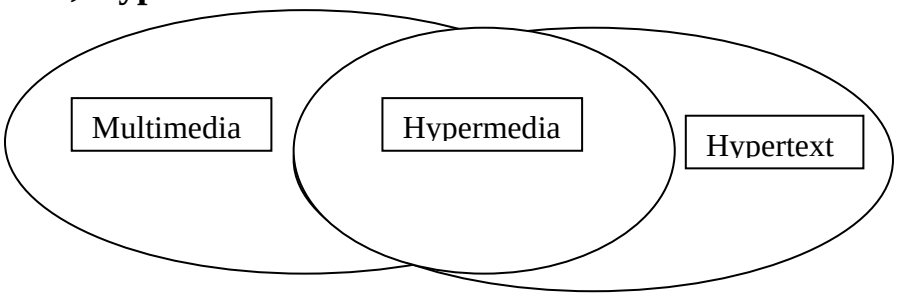

Figure: The hypertext, hypermedia and multimedia relationship.

Hypertext System

A hypertext system is mainly determined through non-linear links of information. Pointers connect the nodes. The data of different nodes can be represented with one or several media types. In a pure text system, only text parts are connected. We understand hypertext as an information object which includes links to several media.

Multimedia System

A multimedia system is characterized by computer-controlled, integrated production, manipulation, presentation, storage and communication of independent information, which is encoded at least through a continuous (time-dependent) and a discrete (time-independent) medium.

For example, if only links to text data are present, then this is not a multimedia system. It is a hypertext.

Hypermedia System

A hypermedia system includes the non-linear information links of hypertext systems and the continuous and discrete media of multimedia systems. For example, if a non-linear link consists of text and video data, then this is a hypermedia, multimedia and hypertext system.

Hypertext System Architecture

The hypertext system architecture can be divided into three layers with different functionalities:

(i) Presentation Layer

All functions connecter to the user interface are embedded. Here, nodes and pointer are mapped to the user interface. At the user interface, one or several parts of the document are visualized. This layer determines, based on the given structure and user's desired display, which data are presented and how they are presented. This layer takes over control of all inputs.

(ii) Hypertext Abstract Machine

The Hypertext Abstract Machine (HAM) is placed between the presentation and storage layers. It can expect from the underlying layer database functions for storage of multimedia data in a distributed environment. It does not have to consider input and output of the upper layer (Presentation Layer). Hypertext Abstract Machine knows the structure of the document, it has the knowledge about the pointers and its attributes. The data structure, respectively a document architecture, is constructed for the management of the document.

(iii) Storage Layer

The storage layer (also called the database layer) is the lowest layer. All functions connected with the storage of data, i.e., secondary storage management, belong to this layer. The specific properties of the different discrete and continuous media need to be considered. Functionalities from traditional database systems are expected, such as persistence (data persist through programs and processes), multi-user operations (synchronization, locks) and the restoration of data after a failure (transaction).

Document Architecture SGML

The Standard Generalized Markup Language (SGML) The Standard Generalized Markup Language, or SGML, is a language and notation for describing classes of documents. In an SGML-encoded document, the various *elements* of the document are delimited with distinguished character strings commonly called *tags*. A document may, for example, have tags that delimit elements like paragraphs, subsections, appendices, and figures. A

start-tag of the form <X> denotes the beginning of an element, and an *end-tag* of the form </X> denotes the end of that element.

An SGML-encoded document has three parts: an *SGML declaration*, a *document type declaration*, and a *document instance*. The SGML declaration characterizes the document type declaration, and the subsequent document instance(s), in terms of character sets and optional features of SGML. The SGML declaration may be omitted if the default character set is used and no optional features are required.

In SGML, a class of documents is characterized by a grammar that indicates what markup is allowed, what markup is required, and how markup is distinguished from text. SGML defines this grammar with a *document type definition.*

SGML document is divided into two processes. Only the formatter knows the meaning of the tag and it transforms the document into a formatted document. The parser uses the tags, occurring in the document, in combination with the corresponding document type. Specification of the document structure is done with tags.

Tags are divided into different categories:

The descriptive markup (tags) describes the actual structure always in the form:

```
<start-tag> respectively also </end-tag>
```

An example is the definition of a paragraph at its beginning:

```
<paragraph> The text of the paragraph follows ......
```

The entity reference provides connection to another element. This element replaces the entity reference. This can be understood also as an abbreviation to which the actual content can be copied later at the corresponding place. The following example shows entity reference in a mathematical context:

```
&square x ....... should be x2
```

The markup declarations define the elements to which an entity reference refers. In our example of squaring a variable x, square is defined as:

```
<!ELEMENT square (....)>
```

A markup declaration can be used to define rules for the structure (the classes). The following example illustrates the construction of an article paper:

```
<!ELEMENT paper (preamble, body, postamble)>
<!ELEMENT preamble (title, author, side)>
<!ELEMENT title (#CDATA)> -- character data
<!ELEMENT body (....)>
```

Instruction for other programs in a text are entered through processing instructions. They can be meant, for example, for the formatter. Using processing instructions, different media can be inserted.

Document Architecture ODA

The Open Document Architecture (ODA) was initially called the Office Document Architecture because it supports mostly office-oriented applications. The main goal of this document architecture is to support the exchange, processing and presentation of documents in open system.

The main property of ODA is the distinction among content, logical structure and layout structure. This is in contrast to SGML where only a logical structure and the contents are defined. ODA also defines semantics. Figure shows these three aspects linked to a document.

(i) Content Portions:

The content of the document consists of Content Portions. These can be manipulated according to the corresponding medium.

A content architecture describes for each medium:

- the specification of the elements,
- the possible access functions and
- the data coding.

Individual elements are the Logical Data Units (LDUs), which are determined for each medium. The access functions serve for the manipulation of individual elements. The coding of the data determines the mapping with respect to bits and bytes.

(ii) Layout Structure:

The layout structure specifies mainly the representation of a document. It is related to a two dimensional representation with respect to a screen or paper.

(iii) Logical Structure:

The logical structure includes the partitioning of the content.

MHEG (Multimedia and Hypermedia Information Coding Expert Group)

The committee ISO/IEC JTC/SC29 (Coding of Audio, Picture, Multimedia and Hypermedia Information) works on the standardization of the exchange format for multimedia systems. The actual standards are developed at the international level in three working groups cooperating with research and industry. Figure shows that the three standards deal with the coding and compression of individual media. The results of the working groups: the Joint Photographic Expert Group (JPEG) and the Motion Picture Expert Group (MPEG) are of special importance in the area of multimedia systems.

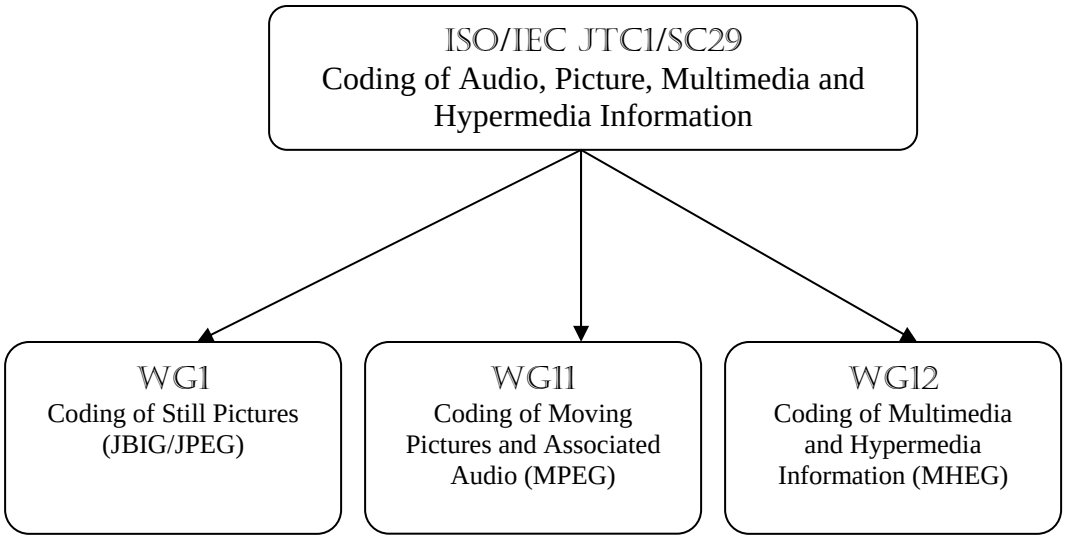

Figure: Working Groups within the ISO-SG29

The standard of this structure description is the subject of the working group WG12, which is known as the Multimedia and Hypermedia Information Coding Expert Group (MHEG). The name of the developed standard is officially called Information Technology-Coding of Multimedia and Hypermedia Information (MHEG). The final MHEG standard will be described in three documents.