

Office document architecture (ODA) and interchange format for text and office systems —

Part 2: Document structures

Committees responsible for this British Standard

The preparation of this British Standard was entrusted by the Information Systems Technology Standards Policy Committee (IST/-) to Technical Committee IST/18, upon which the following bodies were represented:

- British Computer Society
- British Facsimile Industry Consultative Committee
- British Printing Industries Federation
- British Telecommunications plc
- Computing Services Association
- Department of Trade and Industry (Information Technology Division)
- EEA (The Electronics and Business Equipment Association)
- HM Treasury (Central Computer and Telecommunications Agency)
- Her Majesty's Stationery Office
- Joint Network Team
- Ministry of Defence
- National Computer Users' Forum
- National Computing Centre Ltd.
- SGML Users' Group
- Telecommunication Engineering and Manufacturing Association
- University of London
- User Standards Forum for Information Technology (Institute of Data Processing Management)

This British Standard, having been prepared under the direction of the Information Systems Technology Standards Policy Committee, was published under the authority of the Board of BSI and comes into effect on 31 July 1990

© BSI 05-2000

The following BSI references relate to the work on this standard:
Committee reference IST/18
Draft for comment 86/67032 DC

ISBN 0 580 18335 1

Amendments issued since publication

Amd. No.	Date of issue	Comments

Contents

	Page
Committees responsible	Inside front cover
National foreword	v
Section 1. Introduction	
1.1 Scope	1
1.2 Normative references	1
1.3 Definitions, symbols and abbreviations	1
Section 2. Architectural principles	
2.1 Architectural concepts	3
2.2 Structural model of a document	3
2.3 Descriptive representation of a document	5
2.4 Document processing model	10
2.5 Roles of the document architecture constituents in document processing	11
Section 3. Document structures	
3.1 Specific structures	16
3.2 Specific logical structures	17
3.3 Specific layout structure	18
3.4 Documents containing both specific structures	24
3.5 Generic structures	26
Section 4. Interface between the document architecture and content architectures	
4.1 General concept	30
4.2 Specification of a content architecture	30
4.3 Interface information	30
Section 5. Attribute definitions	
5.1 General principles of attributes	32
5.2 Attribute specification format	39
5.3 Shared attributes	39
5.4 Layout attributes	46
5.5 Logical attributes	55
5.6 Layout style attributes	55
5.7 Layout directives	56
5.8 Presentation style attributes	63
5.9 Content portion attributes	64
Section 6. Reference model of the document layout process	
6.1 Overview	66
6.2 Content and layout structure generation	67
6.3 Layout references and categories	68
6.4 The effect of some attributes on the document layout process	70
6.5 Layout process for frames	74
6.6 Allocation of areas for blocks	76
6.7 Alternative representation	78
Section 7. Reference model of the document imaging process	
7.1 Imaging order	79
7.2 Intersection principles	79
7.3 General rules for positioning pages on presentation surfaces	79
Section 8. Document architecture levels	
8.1 Definition of document architecture classes	84

	Page
8.2 The constituents of document architecture classes	84
8.3 Minimum attribute sets	85
Annex A (normative) Notation used to represent document structures	86
Annex B (informative) Examples of document structures	92
Annex C (informative) Examples of document architecture levels	131
Annex D (informative) Examples of particular document architecture features	139
Annex E (informative) The defaulting mechanism	169
Annex F (informative) Attribute summary tables	172
Figure 1 — Views of a document	4
Figure 2 — Descriptive model of a document	9
Figure 3 — Document processing model	12
Figure 4 — Example of sequential order	17
Figure 5 — Permissible specific logical structures	18
Figure 6 — Permissible specific layout structures	20
Figure 7 — The horizontal and vertical directions of layout objects	21
Figure 8 — Naming of the edges of blocks and of their immediately superior frames	22
Figure 9 — Border around a block	23
Figure 10 — Border within a frame	24
Figure 11 — Example of relationship between logical and layout objects and associated content portions	25
Figure 12 — Document architecture — content architecture interface	31
Figure 13 — Illustration of layout streams	71
Figure 14 — Nominal page co-ordinate system and orientation	80
Figure 15 — Dimensions and assured reproduction areas for ISO A4 page size	81
Figure 16 — Dimensions and assured reproduction areas for North American Letter page size	82
Figure 17 — Dimensions and assured reproduction areas for ISO A3 page size	82
Figure 18 — Dimensions and assured reproduction areas for Japanese legal page size	83
Figure 19 — Dimensions and assured reproduction areas for Japanese letter page size	83
Figure A.1 — Example of document structure notation: — main structure	86
Figure A.2 — Example of document structure notation — subtree	87
Figure B.1 — Specimen document “letter” (1)	95
Figure B.2 — Specimen document “letter” (2)	96
Figure B.3 — Specimen document “letter” (3)	97
Figure B.4 — Layout of “letter” showing pages and blocks (1)	98
Figure B.5 — Layout of “letter” showing pages and blocks (2)	99
Figure B.6 — Layout of “letter” showing pages and blocks (3)	100
Figure B.7 — Specific layout structure (showing pages and blocks)	102
Figure B.8 — Specific logical structure	108
Figure B.9 — Generic logical structure	114
Figure B.10 — Generic layout structure	115
Figure B.11 — Specific layout structure	123

	Page
Figure B.12 — Layout structure of “letter” showing “pages”, “frames” and “blocks” (1)	128
Figure B.13 — Layout structure of “letter” showing “pages”, “frames” and “blocks” (2)	129
Figure B.14 — Layout structure of “letter” showing “pages”, “frames” and “blocks” (3)	130
Figure D.1 — Fixed position frame	140
Figure D.2 — Single column text of variable height	141
Figure D.3 — “Wrapping” of text around a left aligned picture	143
Figure D.4 — Horizontally centred picture with text on both sides	146
Figure D.5 — Multi-lingual synchronised text	149
Figure D.6 — Footnote placement	150
Figure D.7 — Tabular layout	152
Figure D.8 — Concatenation of chapter number and title	153
Figure D.9 — Concatenation of parts of a paragraph	153
Figure D.10 — Illustration of attribute “offset”	153
Figure D.11 — Illustration of attribute “separation”	154
Figure D.12 — Layout path 0° and 270°	154
Figure D.13 — Illustration of fill order	155
Figure D.14 — Illustration of block alignment	156
Figure D.15 — Illustration of the attribute “separation”	157
Figure D.16 — Illustration of layout of blocks in normal and reverse order	158
Figure D.17 — Illustration of layout objects in normal order	159
Figure D.18 — Illustration of layout of logical objects in reverse order	160
Figure D.19 — Illustration of synchronization	161
Figure D.20 — Illustration of new layout object	161
Figure D.21 — Illustration of same layout object	161
Figure D.22 — Illustration of indivisibility	162
Figure D.23 — Illustration of balance	162
Figure D.24 — Illustration of overlapping blocks	163
Figure D.25 — Sequential layout order	163
Figure D.26 — Final image	164
Figure D.27 — Sequential layout order	164
Figure D.28 — Image within frame F1	165
Figure D.29 — Imaging of blocks B1, B2, B3	165
Figure D.30 — Image of block B4 added	166
Figure D.31 — Image of block B5 added	166
Figure D.32 — Final image	167
Figure D.33 — Use of bindings and content generator for numbering chapters	168
Table 1 — Defaultable attributes that can be specified in default value lists	47
Table 2 — Document architecture classes	84
Table B.1 — Constituents for Figure B.7, Specific layout structure	103
Table B.2 — Constituents for Figure B.8, Specific logical structure	109
Table B.3 — Presentation styles for Figure B.8	111
Table B.4 — Constituents for Figure B.9, Generic logical structure	115
Table B.5 — Styles for Figure B.9, Generic logical structure	117

	Page
Table B.6 — Constituents for specific logical structure	118
Table B.7 — Constituents for Figure B.10, Generic layout structure	121
Table B.8 — Constituents for Figure B.11, Specific layout structure	124
Table C.1 — Document architecture levels	131
Table C.2 — FDA levels: layout structures	132
Table C.3 — PDA levels: logical structures	133
Table C.4 — PDA levels: applicability of layout directive attributes	134
Table C.5 — PDA levels: layout structures	135
Table C.6 — FPDA levels: logical structures	136
Table C.7 — FPDA levels: applicability of layout directive attributes	137
Table C.8 — FPDA levels: layout structures	138
Table E.1 — Applicable steps for defaulting of attributes	169
Table F.1 — List of attributes of constituents	172
Table F.2 — Attributes which may be specified for constituents shared attributes	174
Table F.3 — Layout directive attributes which may be applied to logical components	176
Publications referred to	Inside back cover

National foreword

This Part of BS 7299 has been prepared under the direction of the Information Systems Technology Standards Policy Committee. It is the UK implementation of, and is identical with, ISO 8613-2:1989 “*Information processing — Text and office systems — Office Document Architecture (ODA) and interchange format — Part 2: Document structures*”, published by the International Organization for Standardization (ISO).

The following list includes all the Parts of this British Standard.

- *Part 1: Introduction and general principles;*
- *Part 2: Document structures;*
- *Part 4: Document profile;*
- *Part 5: Office document interchange format (ODIF);*
- *Part 6: Character content architectures;*
- *Part 7: Raster graphics content architectures;*
- *Part 8: Geometric graphics content architectures.*

NOTE To maintain alignment with the numbering of the Parts of ISO 8613, there is at present no Part 3 of this British Standard.

The UK actively participated in the development of the International Standard through BSI Technical Committee IST/18 and fully approved publication of the International Standard.

There are no plans at present to supplement this British Standard with any additional guidance, and any comments or enquiries on the International Standard or its application should be addressed to the Secretary of IST/18 at the BSI headquarters address.

Reasons for implementation. This Part of BS 7299 implements the International Standard for the following reasons.

- a) Because information technology equipment, services and systems tend to be, and generally need to be, designed, purchased, and used in a world-wide context, BSI policy is to agree, wherever possible, standards at international level and not to prepare unique national standards. At the time of its approval, the text of the International Standard was considered to be the best that could be achieved, taking into account both the needs of the “market place” and the “state of the art” in this subject area, and the international standardization process.
- b) The Commission of the European Communities has mandated the European Committee for Standardization (CEN) to produce European Standards (ENs) [or, where this cannot be achieved, Harmonization Documents (HDs) or Draft European Standards (ENVs)] in this subject area for application throughout the European Community by 1992. This British Standard anticipates that ISO 8613-2:1989 will be adopted without change as the Reference Document for the EN (or HD or ENV) and, therefore, fulfils in advance the UK obligation under CEN/CENELEC Common Rules to implement nationally each agreed European Standard.
- c) Publication of British Standards corresponding to the International Standards in which IST/- resources were invested will ensure that the BSI Catalogue becomes, and then remains, a comprehensive first source of information for UK users of IT equipment and services about the availability of International Standards on IT subjects. In turn, this will facilitate UK use of, and access to, those standards.

Cross-references. The cross-references relating to all Parts of this British Standard are given in the national foreword to Part 1.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, pages i to vi, pages 1 to 176, an inside back cover and a back cover.

This standard has been updated (see copyright date) and may have had amendments incorporated. This will be indicated in the amendment table on the inside front cover.

Section 1. Introduction

1.1 Scope

The purpose of ISO 8613 is to facilitate the interchange of documents.

In the context of ISO 8613, documents are considered to be items such as memoranda, letters, invoices, forms and reports, which may include pictures and tabular material. The content elements used within the documents may include graphic characters, geometric graphics elements and raster graphics elements, all potentially within one document.

NOTE ISO 8613 is designed to allow for extensions, including typographical features, colour, spreadsheets and additional types of content such as sound.

ISO 8613 applies to the interchange of documents by means of data communications or the exchange of storage media.

It provides for the interchange of documents for either or both of the following purposes:

- to allow presentation as intended by the originator;
- to allow processing, such as editing and reformatting.

The composition of a document in interchange can take several forms:

- formatted form, allowing presentation of the document;
- processable form, allowing processing of the document;
- formatted processable form, allowing both presentation and processing.

ISO 8613 also provides for the interchange of ODA information structures used for the processing of interchanged documents.

Furthermore, ISO 8613 allows for the interchange of documents containing one or more different types of content such as character text, images, graphics and sound.

This part of ISO 8613

- defines a document architecture intended for representation of documents;
- defines a document processing model;
- defines the document structures, the basic constituents of the architecture and a descriptive representation of these in terms of attributes;
- defines an interface which allows the use of different content architectures with the document architecture;
- defines the reference model of the document layout process;
- defines the reference model of the document imaging process;

- defines three document architecture classes;
- defines a notation used for illustrating and describing document structures;
- provides examples of document architecture levels;
- provides examples of document structures;
- provides examples of particular document attributes.

1.2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 8613. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 8613 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 6937-2:1983, *Information processing systems — Coded character sets for text communication — Part 2: Latin alphabetic and non-alphabetic graphic characters*.

ISO 8613:1989, *Information processing — Text and office systems — Office Document Architecture (ODA) and Interchange Format —*

- *Part 1: Introduction and general principles;*
- *Part 4: Document profile;*
- *Part 5: Office document interchange format (ODIF);*
- *Part 6: Character content architectures;*
- *Part 7: Raster graphics content architectures.*

ISO 8824:1987, *Information processing systems — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1).*

1.3 Definitions, symbols and abbreviations

1.3.1 Definitions

For the purpose of this part of ISO 8613 the definitions given in ISO 8613-1 apply.

1.3.2 Conventions

The following conventions are used throughout this part of ISO 8613.

1.3.2.1

subordinate

references to the immediate subordinates of an object always use the form “immediate subordinates” or “immediately subordinate”

when the term “subordinate” is used without this qualifier it means subordinate to any hierarchical level

1.3.2.2

superior

references to the immediate superior of an object always use the form “immediate superior” or “immediately superior”

the term “nearest superior” is used in conjunction with an object and with a particular qualification to mean the first superior going up the hierarchy which satisfies the qualification, i.e. first checking the immediate superior, then its immediate superior, and then each superior in turn in this hierarchic order. For example, “nearest superior that belongs to the specified object class” means the first superior going up the hierarchy from the object which is of the specified object class

when the term “superior” is used without either of these qualifiers it means superior to any hierarchical level

1.3.2.3

next and following

when the term “next” or “immediately following” is used in conjunction with an object then it means the object immediately following this object in sequential order (see **3.1.2**). Thus, “next layout object” means the immediately following layout object in sequential layout order; “next logical object” means the immediately following logical object in sequential logical order

when the term “following” is used in conjunction with an object without the qualifier “immediately” then it means an object at any position later in the sequential order than this object

in some cases the term “next” is used together with a further qualification, for example, “next layout object with the same layout category” means the first of the following layout objects in sequential layout order which has the same layout category as the specified layout object

1.3.2.4

preceding

when the term “immediately preceding” is used in conjunction with an object then it means the object immediately preceding this object in sequential order (see **3.1.2**). Thus, “immediately preceding layout object” means the immediately preceding layout object in sequential layout order; “immediately preceding logical object” means the immediately preceding logical object in sequential logical order

when the term “preceding” is used in conjunction with an object without the qualifier “immediately”, then it means an object at any position earlier in sequential order than this object

in some cases the term “preceding” is used together with a further qualification, for example, “preceding layout object with the same layout category” means the last of the preceding layout objects in sequential layout order which has the same layout category as the specified layout object

Section 2. Architectural principles

2.1 Architectural concepts

For the purpose of this International Standard, a document is an amount of structured information that can be interchanged as a unit.

This part of ISO 8613 provides the means to represent the structures of documents in two major forms:

- in a *formatted form* which allows documents to be presented (for example, printed or displayed) as intended by the originator;
- in a *processable form* which allows further processing of documents by the recipient, such as editing and formatting as intended by the originator.

It also provides the means to represent a document in *formatted processable form* to satisfy both purposes.

A document contains information that relates to its content and structure.

The *content* of a document consists of any type of information that is intended for human perception, for example, content elements that can be presented in a two-dimensional form, such as printed on paper or displayed on a screen.

The structural information included in a document is provided in order to

- delimit portions within a document, such as areas for the imaging of different types of content elements;
- delimit portions of a document that have a logical meaning, such as chapters, paragraphs;
- use different types of coding for the different content types;
- allow processing of the document.

The rules for defining the structure and representation of documents are collectively called the *document architecture*.

The document architecture comprises a structural model and a descriptive representation. The *structural model* describes the structural elements of a document and the relationships among these elements. The *descriptive representation* describes how the elements of a document and the properties of these elements are represented.

The structural model and the descriptive representation present complementary views of a document. Both are necessary in order to distinguish between the structural aspects of a document and the data structures which represent those structural aspects. Also, the descriptive representation shows how additional information not described by the structural model is represented. This information includes styles and the document profile. Styles contain information relating to the layout and presentation of a document (see 2.3.5). The document profile includes information relating to the document as a whole (see 2.3.6).

This part of ISO 8613 does not require all parts of the document architecture to be present in any particular document.

2.2 Structural model of a document

The structural model introduces the *structural elements* of the document architecture.

2.2.1 Specific structures

The structural model of a document provides for two different but complementary views of the content of a specific document:

- the logical view associates content with structural elements such as chapters, appendices, headings, paragraphs, footnotes and figures;
- the layout view associates content with structural elements relating to presentation media, such as pages and areas within pages.

This is illustrated in Figure 1.

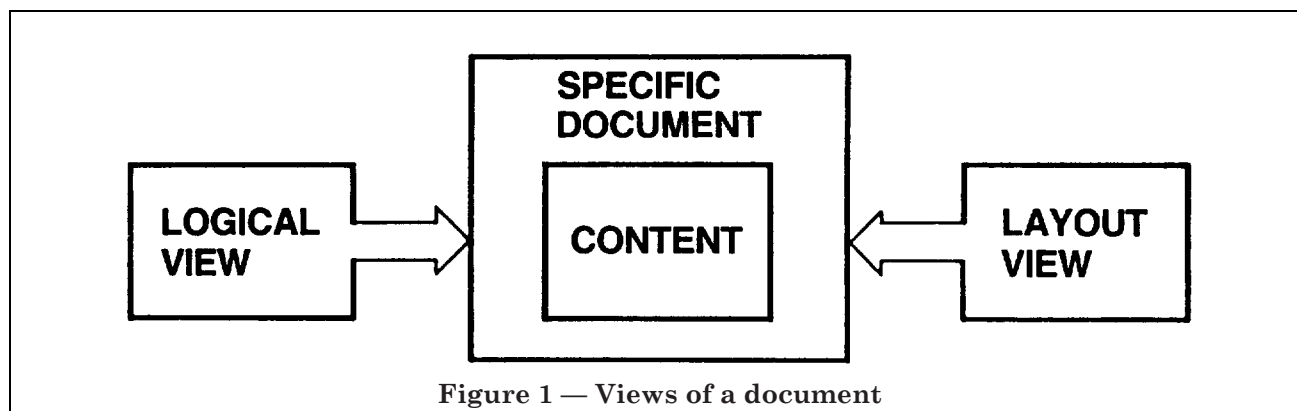
These structural elements of a specific document are called *objects*. Each view associates the same document content with a separate structure which consists of a hierarchy of objects.

Thus

- the *specific logical structure* associates the content of a document with a hierarchy of *logical objects* and provides for the representation of documents in processable form;
- the *specific layout structure* associates the content of a document with a hierarchy of *layout objects* and provides for the representation of documents in formatted form.

The *specific structure* consists of the specific logical structure and/or the specific layout structure.

The structures are independent of the types of content within a document.



There are two types of relationships among objects in a structure, namely

- *structural relationships* which specify the hierarchical structure of the objects;
- *non-hierarchical relationships* which specify other relationships, such as cross-references to figures or footnotes.

2.2.2 Content of a document

The structural model of a document partitions the content into structural elements called *content portions*.

The information within each content portion must pertain to a particular type of content and the structure of this information is defined by a content architecture. A *content architecture* consists of the definition of a set of *content elements*, control functions and attributes, with their coded representation, and of the rules for the application of the attributes and control functions to the content elements.

Selection of content architecture depends on the type or types of content elements to be represented. A single document may contain content portions pertaining to different content architectures. There shall be only one content architecture per content portion.

A content architecture does not identify any logical or layout objects of a document. All structural information and all logical and layout objects are specified by the logical and layout structures of the document architecture.

The document architecture supports the incorporation of the content architectures that are specified in other parts of ISO 8613. The document architecture has been designed to be separate from the content architectures. This is achieved since the structural model has been designed such that the properties of the structural elements that are used to describe a document are independent of the types of content that may be associated with those structural elements. Both document architecture and one or more content architectures are needed to represent a document.

The interface between the document architecture and the content architecture(s) is defined in section 4.

This interface allows the use of any of the content architectures defined in other parts of ISO 8613 with the document architecture.

2.2.3 Generic structures

The generic structure of a document provides for the representation of characteristics common to, and relationships between, a number of objects within a document. In the most comprehensive case it provides for the representation of the common characteristics of a group of documents.

2.2.3.1 Object classes

In many documents, there may be sets of objects with common characteristics; for example

- logical objects representing sections, consisting of a sequence of subordinate objects representing paragraphs, with the same characteristics;
- pages with the same headers and footers.

An *object class* is a structural element of the document that models such a group of common characteristics.

ISO 8613 does not define particular object classes; however it provides the means by which object classes can be defined.

Any content portions associated with an object class are called *generic content portions*.

For convenience of reference, the term *component* is used to refer collectively to an object or an object class.

2.2.3.2 Document classes

A *document class* is used to represent the common characteristics of a group of documents; for example, a set of reports with common paragraphs and common layout. A document class can be used to maintain consistency of an entire document during editing and/or formatting of the document.

The *generic logical structure* provides for the representation of the common characteristics of logical objects of a document class and the *generic layout structure* provides for the representation of the common characteristics of layout objects of a document class.

The generic logical structure consists of all the logical object classes and associated generic content portions of a document.

The generic layout structure consists of all the layout object classes and associated generic content portions of a document.

The *generic structure* consists of the generic logical structure and/or the generic layout structure.

ISO 8613 does not define particular document classes; however it provides the means by which document classes can be defined.

2.3 Descriptive representation of a document

The descriptive representation introduces the descriptive elements of the document architecture.

For the purpose of interchange, a document is represented as a collection of *constituents*, each of which is a set of attributes.

Within ISO 8613 each *attribute* is identified by a name and has a value, which either represents a characteristic of a structural element or a relationship with other constituents.

ISO 8613 defines these constituents and attributes. Those constituents that are counterparts to the structural elements are termed *descriptions*.

2.3.1 Constituents of a document

The following *types of constituent* are defined:

- document profile;
- logical object class description;
- layout object class description;
- logical object description;
- layout object description;
- content portion description;
- presentation style;
- layout style.

Each constituent is characterised by its attributes; within a constituent all attribute names are unique.

2.3.2 Content portion descriptions

Each content portion within a document is characterised by a set of attributes called a *content portion description*.

Any content portion description associated with an object class description (see 2.3.4), for example a logo block or a standard paragraph, is called a *generic content portion description*.

2.3.3 Object descriptions

Each object within a structure is characterised by a set of attributes called an *object description*.

Each attribute has a value and may represent one of the following:

- a) a characteristic of the object itself;
- b) a structural relationship which specifies a hierarchical relationship between the object and other objects in the same structure;
- c) a non-hierarchical relationship between:
 - the object and other objects in the same structure,
 - the object and other objects in different structures included in the same document,
 - the object and object classes included in the same document.

2.3.4 Object class descriptions

Each object class within a document is characterised by a set of attributes called an *object class description*.

An object class description for a logical object class is called a *logical object class description* and an object class description for a layout object class is called a *layout object class description*.

In general, each attribute of an object class description specifies a rule to determine the value of a corresponding attribute of an object description for an object of the object class concerned.

Object class descriptions can be used for the following purposes:

- a) to improve transmission efficiency;
- b) to maintain the internal consistency of a document when it is modified;
- c) to facilitate the creation of objects and documents.

Object class descriptions can be used either individually or collectively.

In either case, the object class descriptions may have generic content portion descriptions associated with them.

In the case of individual use, each object class description is used for transmission efficiency and/or to facilitate creation of objects. Each such object class description consists of a set of attributes representing the common characteristics of objects of the object class.

The set of object class descriptions corresponding to this case of individual use is termed a *factor set* of object class descriptions.

In the case of collective use, two situations must be considered.

In the first situation, some object class descriptions are related to each other in a particular hierarchical structure. This collection serves to facilitate creation of sets of hierarchically related objects within a specific structure, but does not fully specify all possible specific structures that may be created. Such a collection is called a *partial generator set* of object class descriptions.

In the second situation, all object classes are related to each other such that they fully control the generation of specific structures. This collection serves to maintain consistency of an entire document during editing or formatting of the document. During creation and modification of the document, each logical object class description specifies characteristics of the objects that may be created and how these objects may build up the possible specific logical structures of the entire document. Similarly, during document layout, each layout object class description influences the creation of the possible specific layout structures.

The set of object class descriptions corresponding to this second situation is termed a *complete generator set* of object class descriptions.

A generic logical structure or a generic layout structure whose representation includes a partial generator set of object class descriptions is termed a *partial generic logical structure* or a *partial generic layout structure*, respectively.

A generic logical structure or a generic layout structure whose representation includes a complete generator set of object class descriptions is termed a *complete generic logical structure* or a *complete generic layout structure*, respectively.

2.3.5 Styles

In addition to logical and layout component descriptions, a document may contain a number of layout styles and presentation styles which are distinct from the component descriptions.

A style is a set of attributes which may be referred to from component descriptions. The effect of such a reference is to apply the attributes of the style to the component description which contains the reference.

A *layout style* is a set of attributes referred to from one or more logical component descriptions which during document layout guides the creation of a specific layout structure. That is, a layout style provides information for the creation of pages and, if required, guides the division of pages into separate areas, for the layout of the document content.

A *presentation style* is a set of attributes referred to from one or more basic logical and/or layout component descriptions which guides the format and appearance of the content of the document on the presentation medium.

The separation of styles from the component descriptions allows the layout and presentation of a document to be modified without affecting the logical structure.

2.3.6 Document profile

The *document profile* consists of a set of attributes which specify characteristics of the document as a whole.

For instance, the document profile indicates which of the following are present in the document:

- logical object descriptions;
- layout object descriptions;
- logical object class descriptions;
- layout object class descriptions;
- presentation styles;
- layout styles.

The document profile specifies the document architecture class used in the document. It also specifies the ODA version, document application profile, content architectures, interchange format class and document profile level used in the document.

The document profile may describe the document and its history, including information for filing and retrieval and, for example, describing the fonts used in the document.

For the convenience of the recipient, the document profile may duplicate information usually found in the document content (for example, document name, author, date, etc.). However, the document profile does not include data specific to a particular mode of transmission, such as mail, message or Teletex.

A document profile may be interchanged alone to allow a sender to test the capability of the recipient, or for the recipient to get information about the document without sending the complete document.

A complete specification of the document profile is found in ISO 8613-4.

2.3.7 Document class descriptions

A document class is specified by a *document class description*.

A document class description consists of one of the following:

- a) a document profile,
a complete generator set of logical object class descriptions,
optionally layout styles,
optionally presentation styles,
optionally generic content portion descriptions;
or
- b) a document profile,
a complete generator set of layout object class descriptions,
optionally presentation styles,
optionally generic content portion descriptions;
or
- c) a document profile,
a complete generator set of logical object class descriptions,
a complete generator set of layout object class descriptions,
optionally layout styles,
optionally presentation styles,
optionally generic content portion descriptions.

A document class description defines how to generate an entire specific logical and/or specific layout structure for any document of the class.

Documents may reference one externally specified document class description (see 2.3.9). In this case, the document profile indicates whether an external-document class description is referenced by the document, and if so, which one.

2.3.8 Generic-documents

A *generic-document* description consists of one of the following:

- a) a document profile,
a complete generator, partial generator or factor set of logical object class descriptions,
optionally layout styles,
optionally presentation styles,
optionally generic content portions; or
- b) a document profile,
a complete generator, partial generator or factor set of layout object class descriptions,
optionally presentation styles,
optionally generic content portions; or
- c) a document profile,
a complete generator, partial generator or factor set of logical object class descriptions,
a complete generator, partial generator or factor set of layout object class descriptions,
optionally layout styles,
optionally presentation styles,
optionally generic content portions.

A generic-document may be interchanged and used to aid in the generation of documents.

Resource-documents and external-documents are examples of generic-documents.

2.3.9 External-document class description

An interchanged document that does not contain a generic structure can refer to an *external-document*. The external-document is identified in the interchanged document profile. The reference is ignored if any generic layout structure or generic logical structure is present in the interchanged document. Styles may be present in both the interchanged document and the external document.

The external-document can provide any or all of

- constituents representing a complete generic logical structure;
- constituents representing a complete generic layout structure;

and optionally

- layout styles;
- presentation styles.

An external-document contains a document profile which supplies information for the constituents and styles in the external-document, for example, a fonts list. The information in this document profile cannot be used by the interchanged document except by reference to the constituents and styles in the external-document.

If layout or presentation styles in the interchanged document and the external-document have the same identifier then the style in the interchanged document is used and the style in the external-document is ignored.

2.3.10 Resource-documents

An object class description in a given interchanged document may contain a reference to an object class description external to the document, in a generic-document. The latter generic-document is called the *resource-document* of the interchanged document.

The object class descriptions contained in the resource-document act as models for object class descriptions in the interchanged document.

The attributes constituting an object class description in the resource-document may supply values for the corresponding attributes of those object class descriptions in the interchanged document that refer to this object class description in the resource-document.

A resource-document may contain generic content portion descriptions to be included by reference into an interchanged document referring to this resource-document.

Thus, the relationship between an object class description in the interchanged document and the corresponding object class description in the resource-document is similar to the relationship between an object description and the corresponding object class description in the interchanged document.

A resource-document is separate from any document or documents referring to it.

2.3.11 Classes of document architecture

A *document architecture class* is a set of rules for defining the structure and representation of documents in formatted form, processable form, or formatted processable form.

Section 8 defines three document architecture classes that may be used in conjunction with document application profiles, as defined in ISO 8613-1.

The three classes of document architecture are

- a) *Formatted document architecture class* which allows for document content to be presented as intended by the originator; for example, printed or displayed. A document of this class includes a document profile and constituents representing a specific layout structure. It may also include constituents representing a generic layout structure and presentation styles.

- b) *Processable document architecture class* which allows for document content to be processed; for example, edited or formatted. A document of this class includes a document profile and constituents representing a specific logical structure. It may also include constituents representing a generic logical structure, a generic layout structure, layout styles and presentation styles.

- c) *Formatted processable document architecture class* which allows for document content to be processed as well as presented as intended by the originator. A document of this class includes a document profile, constituents representing a specific logical structure, a specific layout structure and a generic layout structure. The generic layout structure can be omitted if there is a reference to an external-document containing at least a complete generic layout structure. The document may also include constituents representing a generic logical structure, layout styles and presentation styles.

A generic-document may be assigned to one of the preceding classes, by the following rules:

- a) If the generic-document contains logical object class descriptions, and no layout object class descriptions, it is assigned to the processable document architecture class;
- b) If the generic-document contains layout object class descriptions, and no logical object class descriptions, it is assigned to the formatted document architecture class;
- c) If the generic-document contains both logical and layout object class descriptions, it is assigned to the formatted processable document architecture class.

2.3.12 Sets of constituents

A document is represented by constituents, which are grouped into sets of constituents, and which have inter-relationships, as defined in ISO 8613.

The possible types of constituent in the descriptive representation of a document are shown in Figure 2.

Figure 2 illustrates that

- a) the document consists of one document profile and optionally a number of constituents that form the document body;

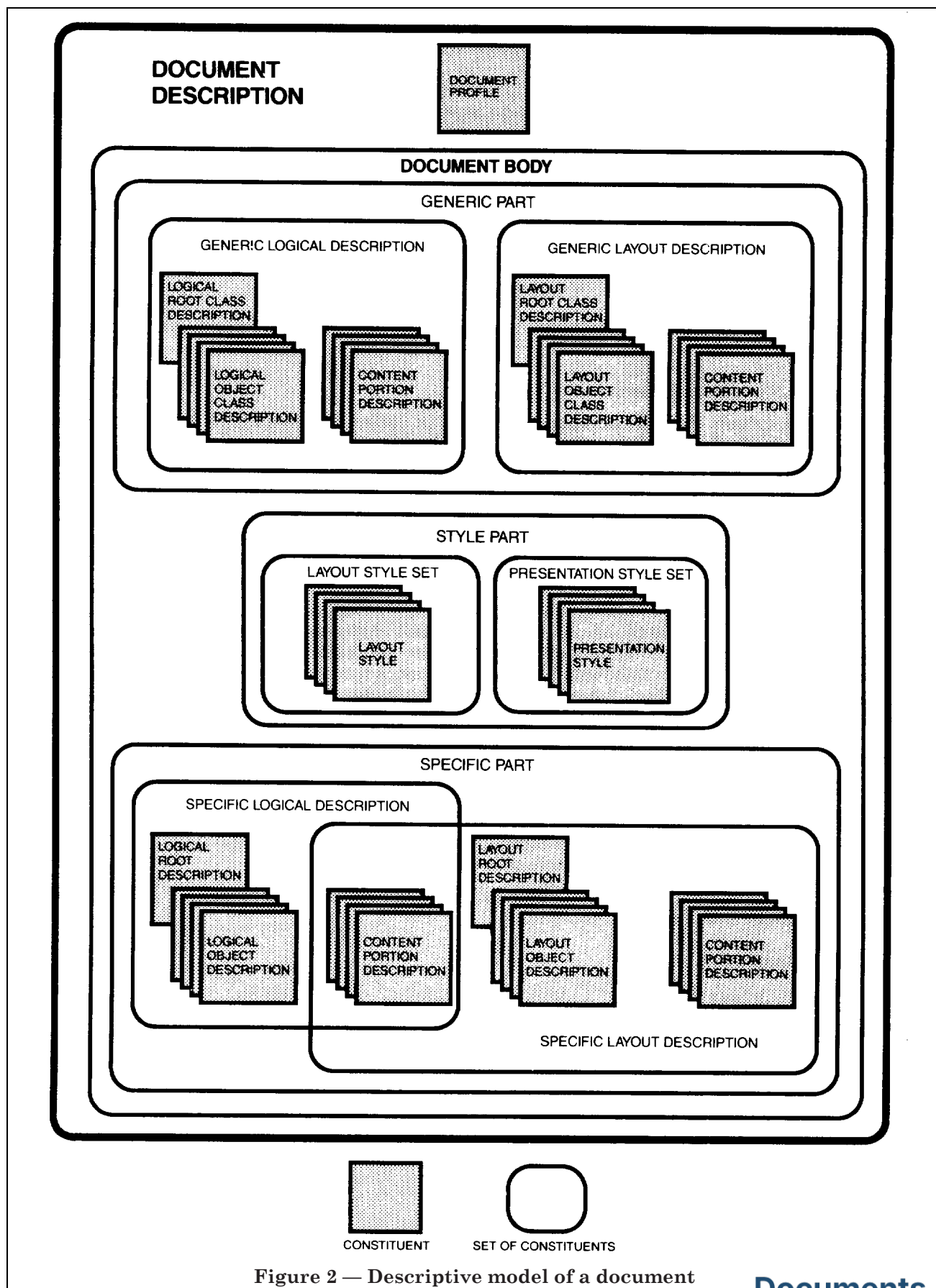


Figure 2 — Descriptive model of a document

b) the *document body* consists of one of the following sets:

- 1) constituents representing the generic structure, and optionally style constituents;
- 2) constituents representing the specific structure, and optionally style constituents;
- 3) constituents representing the generic structure, the specific structure, and optionally style constituents;

c) the constituents representing the generic structure consist of constituents representing the generic logical structure and/or constituents representing the generic layout structure;

d) the style constituents of a document consist of layout styles and/or presentation styles;

e) the constituents representing the specific structure consist of constituents representing the specific logical structure and/or constituents representing the specific layout structure;

f) the constituents representing the generic logical structure consist of the logical object class descriptions and any associated generic content portion descriptions;

g) the constituents representing the generic layout structure consist of the layout object class descriptions and any associated generic content portion descriptions;

h) the constituents representing the specific logical structure consist of the logical object descriptions and any associated content portion descriptions;

j) the constituents representing the specific layout structure consist of the layout object descriptions and any associated content portion descriptions;

k) if both the specific logical structure and specific layout structure are present in a document, then the content portions associated with these structures are, in general, common to both.

2.4 Document processing model

This clause describes a conceptual model for document processing.

It addresses only those aspects of document processing that are relevant to the document architecture defined in ISO 8613. As such, it is not a complete model of document processing since it does not specify all processing steps from document creation to document imaging.

It describes the principal operations performed on a document as a basis for understanding the semantics of the attributes defined in section 5.

It is not intended to represent an actual implementation, nor to restrict in any way the processing that may be applied to an interchanged document.

The document processing model (see Figure 3) is summarized in this section. Three processes are shown:

- the editing process;
- the layout process;
- the imaging process.

The order of processes in the diagram is not intended to imply that they are performed sequentially in an actual implementation.

The document processing model provides for manual intervention only for carrying out editing processes on the specific logical structure and content. Manual intervention can occur, however, at many stages in the model, but it is neither explicitly allowed nor prohibited by ISO 8613.

For example, in an actual implementation it may be possible to create and amend the logical and layout object class descriptions, and layout and presentation styles, but these processes are not included in this model since ISO 8613 does not place any constraints on such editing.

2.4.1 The editing process

The editing process includes both document creation and document revision as these activities are indistinguishable from an architectural perspective.

This *editing process* includes the *content editing process* and the *logical structure editing process*. The content editing process is concerned with the creation of new content or the modification of previous content. The logical structure editing process is concerned with the creation of a specific logical structure or the modification of a previous specific logical structure and the allocation of content to basic logical objects. Modifications to the specific logical structure are required to conform to the rules specified in the generic logical structure, if present.

Included in the document creation and document revision activities are the creation of the generic logical structure, the generic layout structure, layout styles and presentation styles. Styles may be altered to represent changes made during the logical structure editing process. The generic layout structure may be edited in order to alter the intended layout of the document.

2.4.2 The layout process

The *layout process* includes the *document layout process* and the *content layout process*. These processes are concerned with the creation of a specific layout structure which can be used by the *imaging process* to present the document in human perceptible form on a presentation medium.

The document layout process creates a specific layout structure in accordance with the generic layout structure and information derived from the specific logical structure, the generic logical structure and layout styles (if present).

This process also determines the areas that are available within the created layout objects for the formatting of the document content (as described below) and is responsible for allocating the content to these *available areas*.

The content layout process is responsible for formatting (or laying out) the content portions into the available areas specified by the document layout process. This process makes use of information contained in the presentation attributes that apply to those content portions.

During the layout process, presentation attributes may be derived from presentation styles referenced by the logical structure, from the generic layout structure, and from presentation styles referenced by the generic logical structure. Derivation of presentation attributes from the logical structures has precedence over derivation from the generic layout structure. Those presentation attributes derived from the logical structures that differ from those derived from the generic layout structure will be specified explicitly in the specific layout structure by the layout process.

The document layout process is described in section 6. The content layout process depends on the content architecture pertaining to the content being laid out and is described in those parts of ISO 8613 that are concerned with particular content architectures.

The role of the constituents that represent the generic logical structure is different in the editing process and the layout process. In the editing process, the logical object class descriptions are used to construct the specific logical structure. In the layout process, a logical object class description is used as a source of attributes and content, if any, that is common to the logical objects of the class.

2.4.3 The imaging process

The *imaging process* consists of taking a specific layout structure and a corresponding generic layout structure (if present), with associated formatted content portions and information contained in presentation styles, and displaying it on a suitable presentation medium.

Some aspects particular to the imaging process are described in section 7. However, the imaging process is a locally defined process. Hence, apart from defining the input information required by this process, which is the specific and generic layout structures, referenced presentation styles and the formatted content portions, this process is not formally defined in ISO 8613.

During the imaging process, presentation attributes are derived from the layout structures and referenced presentation styles. Any presentation attributes specified by the logical structures and referenced presentation styles are ignored.

The role of the constituents that represent the generic layout structure is different in the document layout process from that in the imaging process. In the document layout process, the layout object class descriptions are used to construct the specific layout structure. In the imaging process, a layout object class description is used as a source of attributes and content, if any, that is common to the layout objects of the class.

Some aspects of imaging depend on the content architecture pertaining to the content to be imaged and are described in those parts of ISO 8613 that are concerned with particular content architectures.

2.5 Roles of the document architecture constituents in document processing

2.5.1 Editing process

This clause describes the role of the various parts of the document architecture when the editing process is applied to documents of processable or formatted processable class.

2.5.1.1 Generic logical structure

The generic logical structure can be used to control the editing process that can be carried out on a document. This is accomplished by providing references from logical object descriptions to logical object class descriptions.

If a complete generator set of logical object class descriptions is present in the document then there is a reference from every logical object description to a logical object class description.

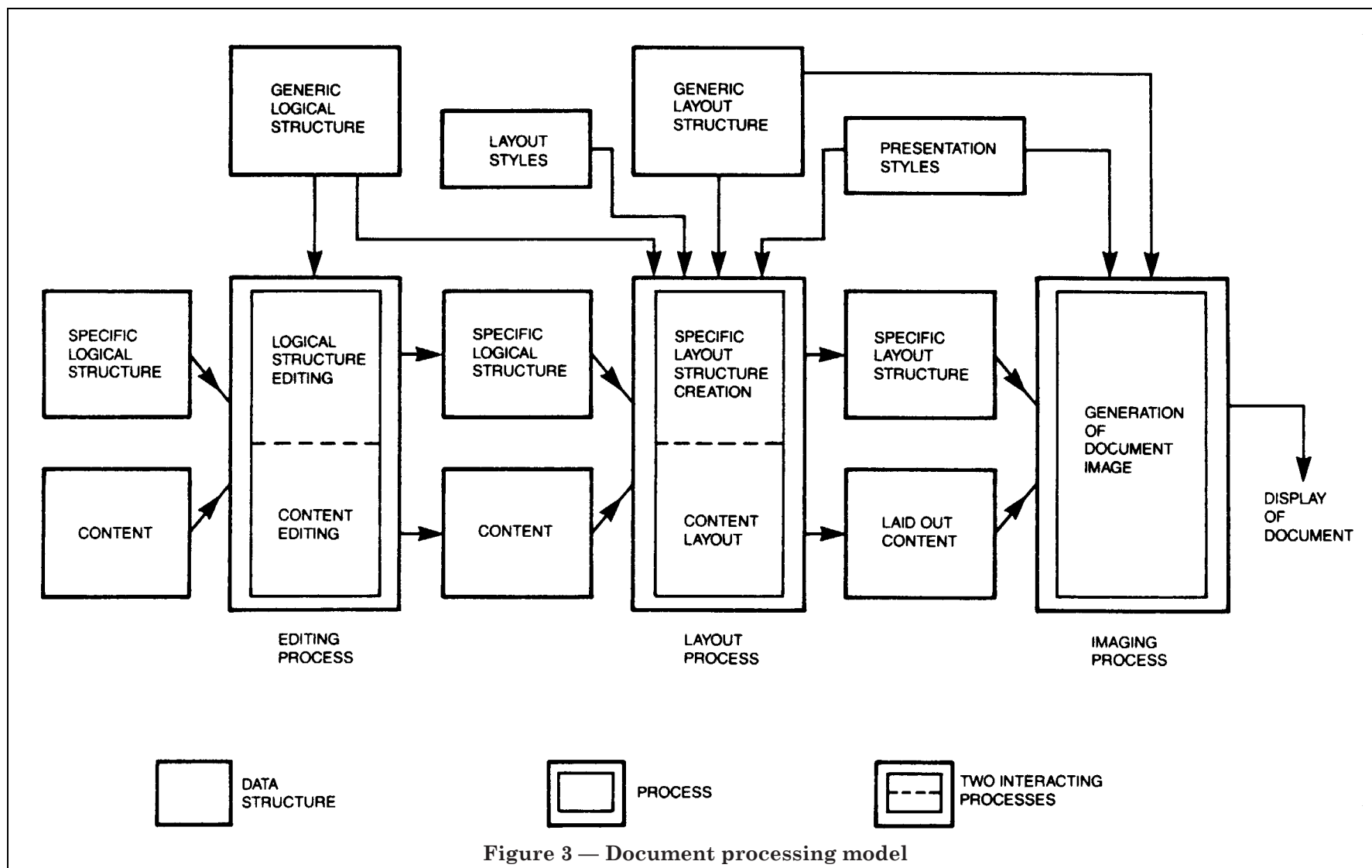


Figure 3 — Document processing model

A complete generator set of logical object class descriptions controls the specific logical structure which is produced during the document creation and editing process. This is achieved by ensuring that logical objects are only created, deleted or modified according to corresponding logical object classes in the generic logical structure. In addition, the creation of a new logical object is facilitated because a logical object class description serves as a template for the logical object description.

A partial generator set of logical object class descriptions is similar to a complete generator set, except that it controls only portions of the specific logical structure rather than all of it.

If a factor set of logical object class descriptions is present in the document, then the object classes in that structure can be used to facilitate the creation of objects in the specific logical structure that have common properties. This is achieved by providing references from a logical object description to a logical object class description. This provides a means of reducing the amount of common information in objects of the specific logical structure, this is referred to as “factorization” of information.

A factor set of logical object class descriptions has no control over the structure of the specific logical structure.

2.5.1.2 Specific logical structure

The logical structure editing process involves making changes in the specific logical structure. These changes may be as follows:

- creating or deleting a logical object;
- changing the position of an object in the specific logical structure;
- changing the characteristics of an object by adding, deleting or modifying attributes specified by the object description.

Changing the characteristics of an object can also involve

- changing the layout styles referenced by the object description;
- changing, adding or deleting a default value for an attribute in a default value list for a logical object description.

2.5.1.3 Content

The content editing process involves making changes to the content. Representing this can involve making changes to the presentation styles applicable to the basic logical object descriptions.

In order to edit the content it must be in processable or formatted processable form. The content is changed by

- a) adding, modifying or deleting one or more content elements;
- b) adding, modifying or deleting embedded control functions.

Editing of formatted form content is outside the scope of ISO 8613. However, all three forms of content may be specified by the content portions of the logical structure.

The editing algorithms used to change the content are not described in this model as they are outside the scope of ISO 8613.

2.5.1.4 Generic layout structure

The generic layout structure, if present in the document, plays no direct role in the editing process.

However, the generic layout structure may be edited in order to alter the intended layout of the document. This process is of local concern only and is outside the scope of ISO 8613.

2.5.1.5 Specific layout structure

The specific layout structure, if present in the document, plays no direct role in the editing process.

However, the specific layout structure which will be produced by the document layout process may be changed as a consequence of the editing process modifying the specific logical structure or the document content.

2.5.1.6 Layout styles

Layout styles are not used to represent controls on the editing process.

Layout styles may be altered to represent changes made during the logical structure editing process (see 2.5.1.2). This involves the addition, deletion or modification of the layout directive attributes contained in a layout style. This can affect the layout characteristics of all the logical objects whose representation refers to the style and can affect the relationships that the objects have with other objects in the document. The result of changing the layout styles is that it may cause changes in the specific layout structure produced by the document layout process.

2.5.1.7 Presentation styles

Presentation styles are not used to represent controls on the editing process.

Presentation styles may be altered to represent changes made during the content editing process. This involves the addition, deletion and modification of attributes contained in the presentation styles. This can affect the layout and imaging of the content associated with all the basic logical objects whose representation refers to the presentation styles that are changed. For some presentation attributes, the same effects may be produced by editing the control functions that are embedded in the content but such editing will only alter the layout and imaging of the content in which the edited control functions are embedded.

2.5.2 Layout process

This clause describes the role of the various parts of the document architecture when the layout process is applied to documents of processable or formatted processable class. A model of this process is described in section 6 of this part of ISO 8613.

2.5.2.1 Generic logical structure

During the layout process, the generic logical structure may provide layout styles, presentation styles and generic content portions which may affect the creation of the specific layout structure.

Also, object classes may be referenced by the attribute “logical source” that is contained within layout object class descriptions. This results in the creation of additional layout objects that have no correspondence with any objects in the specific logical structure.

2.5.2.2 Specific logical structure

During the layout process, the sequential logical order of the objects in the specific logical structure determines the sequence in which the content of the document is considered by the layout process.

In addition, the logical objects can provide layout directives which direct the document layout process and presentation attributes which direct the content layout process. These layout directives and presentation attributes are specified by referencing a layout style or a presentation style, respectively.

2.5.2.3 Layout styles

During the layout process, layout styles provide layout directives which direct the document layout process.

2.5.2.4 Presentation styles

During the layout process, presentation styles provide presentation attributes which direct the content layout process.

2.5.2.5 Content

During the layout process, the content is allocated to basic layout objects. The division of the content into content portions may be modified so that it is consistent with both specific structures. In addition, the content layout process may insert control functions into the content to facilitate the imaging process.

2.5.2.6 Generic layout structure

During the layout process, a complete generator set of layout object class descriptions must be available to determine a specific layout structure for the document.

The construction expressions specified by the layout object class descriptions determine all permissible specific layout structures which may be created by the layout process. Which of these permissible structures is used is determined from the specific logical structure, the generic logical structure, the content, and the layout and presentation styles.

2.5.2.7 Specific layout structure

The specific layout structure results from applying the document and content layout processes to the specific logical structure, the content, the generic logical structure, the generic layout structure, layout styles and presentation styles.

2.5.2.8 Generic content

Generic content portions associated with logical object classes may be in formatted, processable or formatted processable form. Whenever such a logical object class specifying a generic content portion is referred to from a basic logical object in a specific logical structure, the generic content is laid out in accordance with the document and content layout processes and a new basic layout object is created and added together with the created content portion to the specific layout structure.

Generic content portions associated with layout object classes may be in formatted or formatted processable form. The layout of such content portions during the layout process does not cause additional content portions to be added to the specific layout structure. However, the content portions are indirectly associated with the specific layout structure by reference to the appropriate layout object class description.

2.5.3 Imaging process

This clause describes the role of the various parts of the document architecture when the imaging process is applied to documents of formatted or formatted processable class.

2.5.3.1 Content

In the imaging process, the content, together with the specific layout structure, is used to produce an image of the document perceptible to a human.

After layout, the content may be either in formatted form or in formatted processable form. Both forms of content are suitable for imaging.

2.5.3.2 Generic layout structure

During the imaging process, the generic layout structure may provide, for the layout objects in the specific layout structure, any combination of the following:

- a) attributes that direct the imaging of the content;
- b) generic content portions.

2.5.3.3 Specific layout structure

In the imaging process, the specific layout structure, together with the content, is used to produce a human-perceptible image of the document.

The sequential order of the objects in the specific layout structure determines the precedence for imaging the content of the document by the imaging process. It is also possible to change the precedence of the layout objects by specifying an imaging order which is different from the sequential layout order.

2.5.3.4 Layout styles

In the imaging process, layout styles play no role.

2.5.3.5 Presentation styles

In the imaging process, the presentation attributes of the presentation styles referenced by the layout structures may affect the image generation of the content.

Section 3. Document structures

3.1 Specific structures

3.1.1 General principles

The specific layout and specific logical structures of a document are hierarchical structures of objects.

The object at the highest level in the hierarchy of the specific layout structure is called the document layout root and the object at the highest level in the hierarchy of the specific logical structure is called the document logical root.

A *composite object* is an object that has one or more subordinate objects. The structural relationships of a composite object identify its immediate subordinates.

The minimum number of hierarchical levels below the highest level in either structure is one. Thus, the document root object is always a composite object. The actual number of levels is variable and depends upon a given document.

In general, at any level in a specific structure, the subordinates of any composite object can consist of any number and combination of basic objects and composite objects.

Basic objects are objects that have no subordinate objects.

Basic objects are also distinguished from composite objects since basic objects are the only objects with which the content of a document is directly associated (see 3.1.3). Every basic object has content, either in the form of one or more specific content portions, in the form of an expression for generating content or derived from an object class to which the basic object belongs.

Each object in a structure is of a certain object type. The object types that can occur within a specific logical structure or a specific layout structure are defined in 3.2 and 3.3, respectively. The *object type* determines which attributes are applicable to an object description.

The particular attributes and attribute values comprising the object description characterise the object, that is, the attributes specify the characteristics of the object itself and specify the relationships that it has with other components in the document.

Each object in a structure is uniquely identified within that structure. The structural relationships between a composite object and its immediate subordinates are defined in the composite object description.

3.1.2 Sequential order

In the case that more than one immediate subordinate is identified by a composite object then the composite object description specifies an ordering of these immediately subordinate objects. This subordinate ordering is used to define a unique sequential ordering of all the objects in a structure.

This sequential order is such that each object in the structure is succeeded by all of its immediate subordinates, before any other objects with the same immediate superior. Each of the immediate subordinates is followed by all of its immediate subordinates, before proceeding to the next immediate subordinate in sequence. The immediate subordinates occur in the subordinate ordering defined within the object description.

The sequential order of the specific logical structure is called the sequential logical order; the sequential order of the specific layout structure is called the sequential layout order.

The *sequential logical order* defines the order that the logical objects are intended to be processed by the layout process (see section 6).

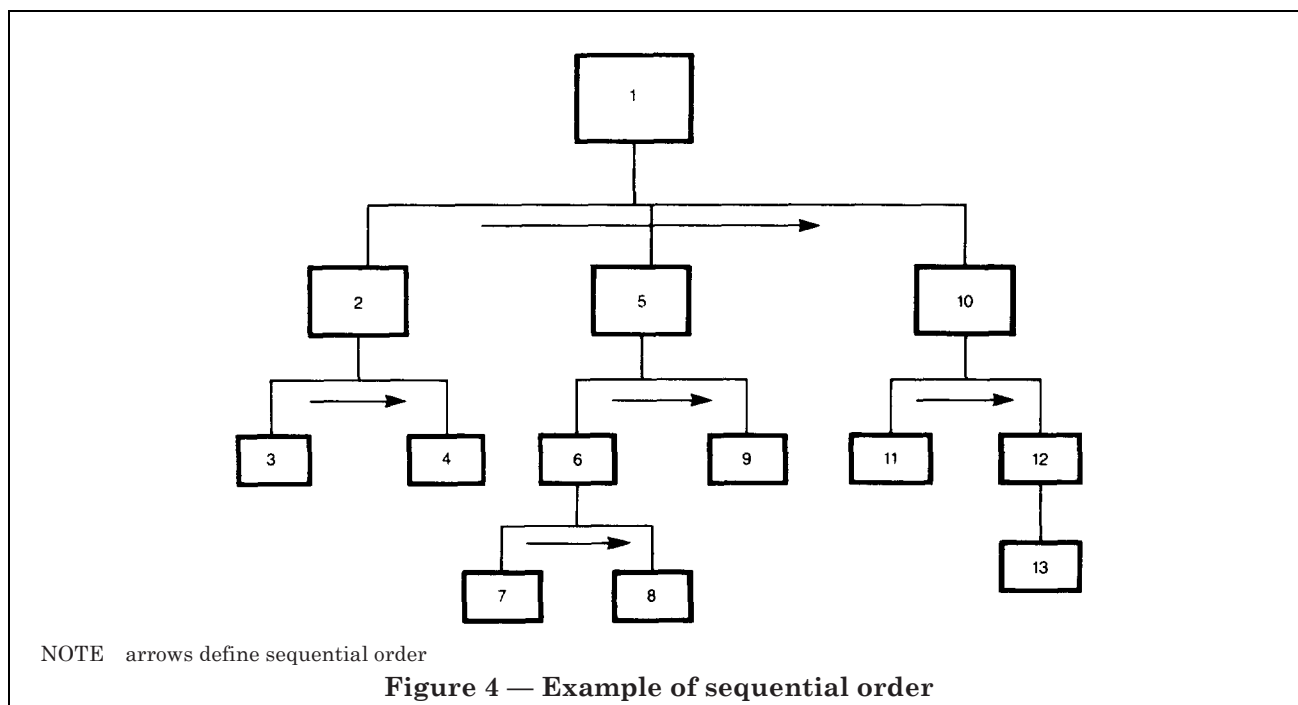
The *sequential layout order* defines the order of precedence for imaging unless this is overridden by the specification of an imaging order in the object description (see 7.1).

An example of the sequential order of a structure is shown in Figure 4, where the numbers indicate the sequential order.

3.1.3 Content portions

The content of a document is divided into content portions to allow the document architecture to address each content portion as a unit. The amount of content to be placed into a content portion may vary from no content elements to the entire document content. The maximum number of consecutive content elements that have the same properties and characteristics will typically be placed together in one content portion. For example, a content portion may represent a heading, paragraph, picture or perhaps just that amount of content that needs to have particular constraints regarding its imaging or processing.

Each content portion is structured according to a single content architecture. When a document contains only a single specific structure, either a specific layout structure or a specific logical structure, each content portion in the specific structure is associated with a single basic object.



Each basic object may have more than one associated content portion, provided that all of these content portions pertain to the same content architecture. In the case that more than one content portion is associated with a basic object then the object description specifies the ordering among these content portions.

3.2 Specific logical structures

The specific logical structure provides a means of modelling the structure of a document in terms of logical objects which have meanings that are significant to the application or user.

For example, the specific logical structure can be used to model a document in terms of chapters, sections, paragraphs and footnotes. ISO 8613 does not standardise such particular kinds of logical objects. However, it does provide the means by which any such construction can be defined in terms of logical objects to be interpreted as chapters, sections, paragraphs or footnotes etc.

The significance of this form of definition is that instead of standardising a few particular kinds of logical objects it allows any number of differing kinds of logical objects to be constructed, so as to meet a wide variety of needs.

Logical objects provide the means to specify the specific logical structure of a document that is relevant to a particular application of ISO 8613. For example, the specific logical structure may be required in the processing of a document, such as editing and laying out the document.

Section 5 defines the attributes that are used to describe the characteristics of logical objects, including the use of presentation styles and layout styles.

The objects that can occur within a specific logical structure of a document are of the following object types:

- document logical root;
- composite logical object;
- basic logical object.

The allowable hierarchical relationships between logical objects are shown in Figure 5.

NOTE The notation used in this figure is that defined in Annex A.

3.2.1 Document logical root

The *document logical root* is the highest level object in the hierarchy of the specific logical structure. It is a composite object whose immediate subordinates consist of any number and combination of basic logical objects and composite logical objects.

3.2.2 Composite logical objects

A *composite logical object* is a composite object that is immediately subordinate either to the document logical root or to a composite logical object of a hierarchically higher level. Its immediate subordinates consist of any number and combination of composite logical objects and basic logical objects. Thus, the number of hierarchical levels between any given basic logical object and the document logical root is variable.

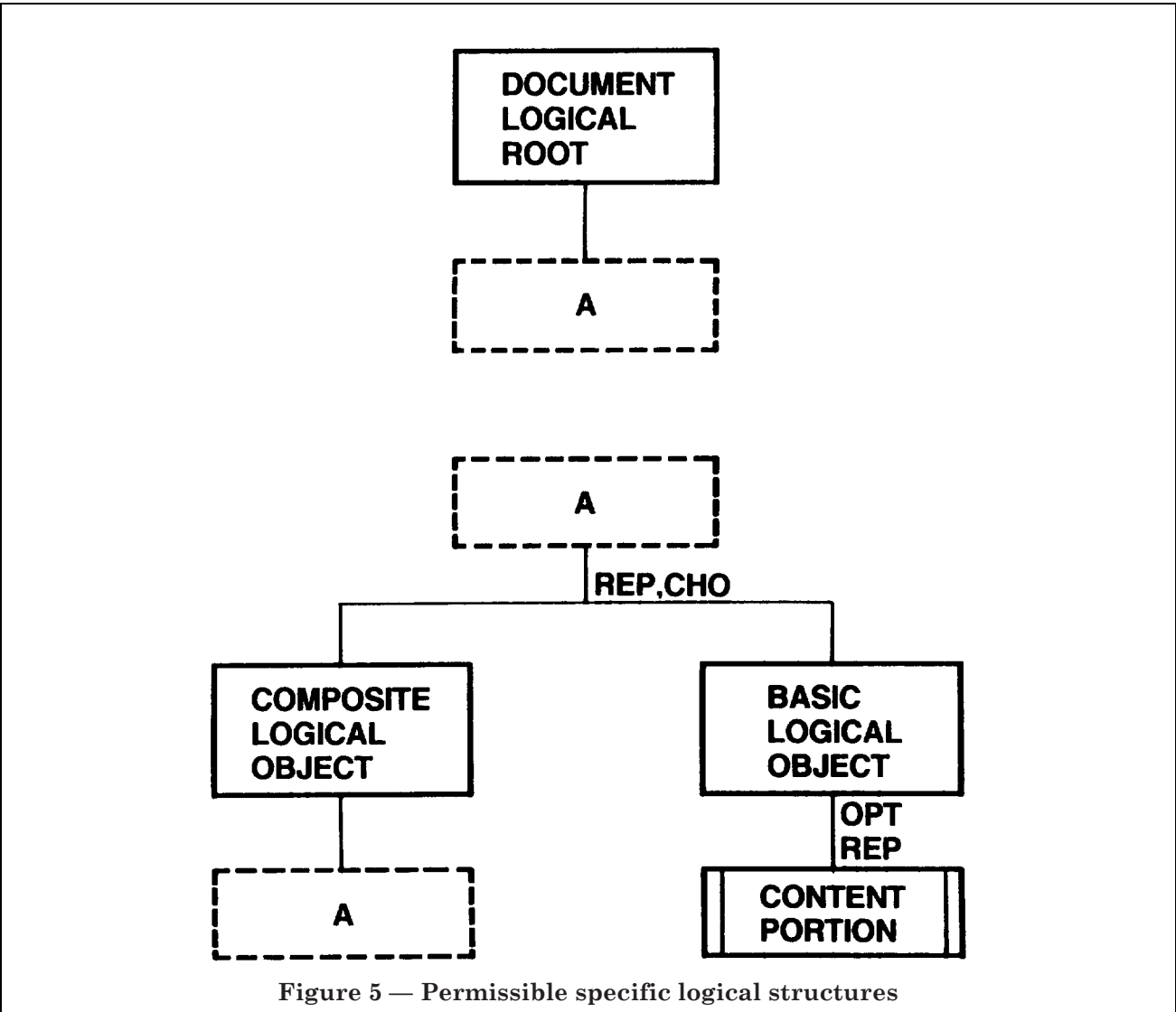


Figure 5 — Permissible specific logical structures

The use of composite objects is optional. Content portions cannot be directly associated with composite logical objects.

The hierarchical relationships between a composite logical object and its subordinates can express a logical relationship between that composite logical object and its subordinates that has a significance relevant to a particular application of ISO 8613. For example, these relationships may be used to specify which sections, paragraphs and diagrams relate to a certain chapter in a document.

3.2.3 Basic logical objects

A *basic logical object* is a basic object that is immediately subordinate either to the document logical root or to a composite logical object.

A basic logical object is a container for portions of the document content. Zero, one or more content portions are directly associated with a basic logical object. In the case that zero content portions are directly associated, the content is either specified in the form of an expression for generating content or is derived from a basic object class description.

3.3 Specific layout structure

The specific layout structure provides a means of modelling the structure of a document in terms of layout objects which have meanings that are significant for the layout process and the imaging process.

Section 5 defines the attributes that are used to describe the characteristics of layout objects, including the use of presentation styles.

3.3.1 Objects of the layout structure

The objects that can occur within the specific layout structure of a document are of the following object types:

- document layout root;
- page set;
- composite or basic page;
- frame;
- block.

The allowable hierarchical relationships between layout objects are shown in Figure 6.

NOTE The notation used in this figure is that defined in Annex A.

3.3.1.1 Document layout root

The *document layout root* is the highest level object in the hierarchy of the specific layout structure. It is a composite object whose immediate subordinates consist of any number and combination of page sets and pages.

3.3.1.2 Page set

A page set can be used to identify a number of page sets or pages (or a combination of both) as a group, for example, the pages which contain a particular section of a document.

A *page set* is a composite object that is immediately subordinate to either the document layout root or to a page set of a hierarchically higher level. Its immediate subordinates consist of any number and combination of page sets and pages. Thus the number of hierarchical levels between any given page and the document layout root is variable.

3.3.1.3 Page

A *page* is a rectangular area used as the reference area for positioning and imaging the content of the document. Its size may be smaller than, equal to or greater than the size of the presentation medium.

A page is immediately subordinate either to the document layout root or to a page set.

A page can be a basic object or a composite object. The pages within a layout structure can either be basic or composite pages, but not a combination of these.

If a page is a basic object, then zero, one or more content portions are directly associated with the page. In the case that zero content portions are directly associated, the content is derived from a basic object class description.

If a page is a composite object, then its immediate subordinates consist of either any number of frames or any number of blocks.

3.3.1.4 Frame

A *frame* is a rectangular area that is contained entirely within the area of the object to which it is immediately subordinate. It is positioned so that its sides are parallel to the sides of the enclosing page. Frames are used to define areas for laying out the content associated with composite pages.

A frame is a composite object that is immediately subordinate to a composite page or to an enclosing frame. The immediate subordinates of a frame consist of either any number of frames or any number of blocks. A *lowest level frame* on any branch of the layout structure is a frame which does not have any subordinate frames. Only lowest level frames can contain blocks. The number of hierarchical levels between any given block and the enclosing page is variable.

3.3.1.5 Block

A *block* is a rectangular area that is contained entirely within the area of the object to which it is immediately subordinate. It is positioned such that its sides are parallel to the sides of the enclosing page. A block is an area for the positioning and imaging of portions of the document content.

A block is a basic object that is immediately subordinate to a page or a frame. Zero, one or more content portions are directly associated with a block. In the case that zero content portions are directly associated, the content is derived from a basic object class description.

3.3.2 Positioning layout objects

This clause defines the rules for positioning and dimensioning frames and blocks within pages. These layout objects are used to layout and image the content of documents.

3.3.2.1 Page co-ordinate system

The positions of all layout objects subordinate to pages are specified directly or indirectly, by means of an orthogonal *page co-ordinate system*. The origin of the page co-ordinate system is the top left corner of the page. The horizontal axis coincides with the top edge and the vertical axis coincides with the left edge of the page. The horizontal and vertical axes determine the *horizontal direction* and *vertical direction*, respectively, of the subordinate layout objects, as shown in Figure 7. This figure also defines names for corners of a layout object: *top left corner*, *top right corner*, *bottom left corner* and *bottom right corner*. Horizontal positions are measured positively from the vertical axis to the right and vertical positions are measured positively from the horizontal axis downward.

The origin of the page co-ordinate system is the reference point used for positioning.

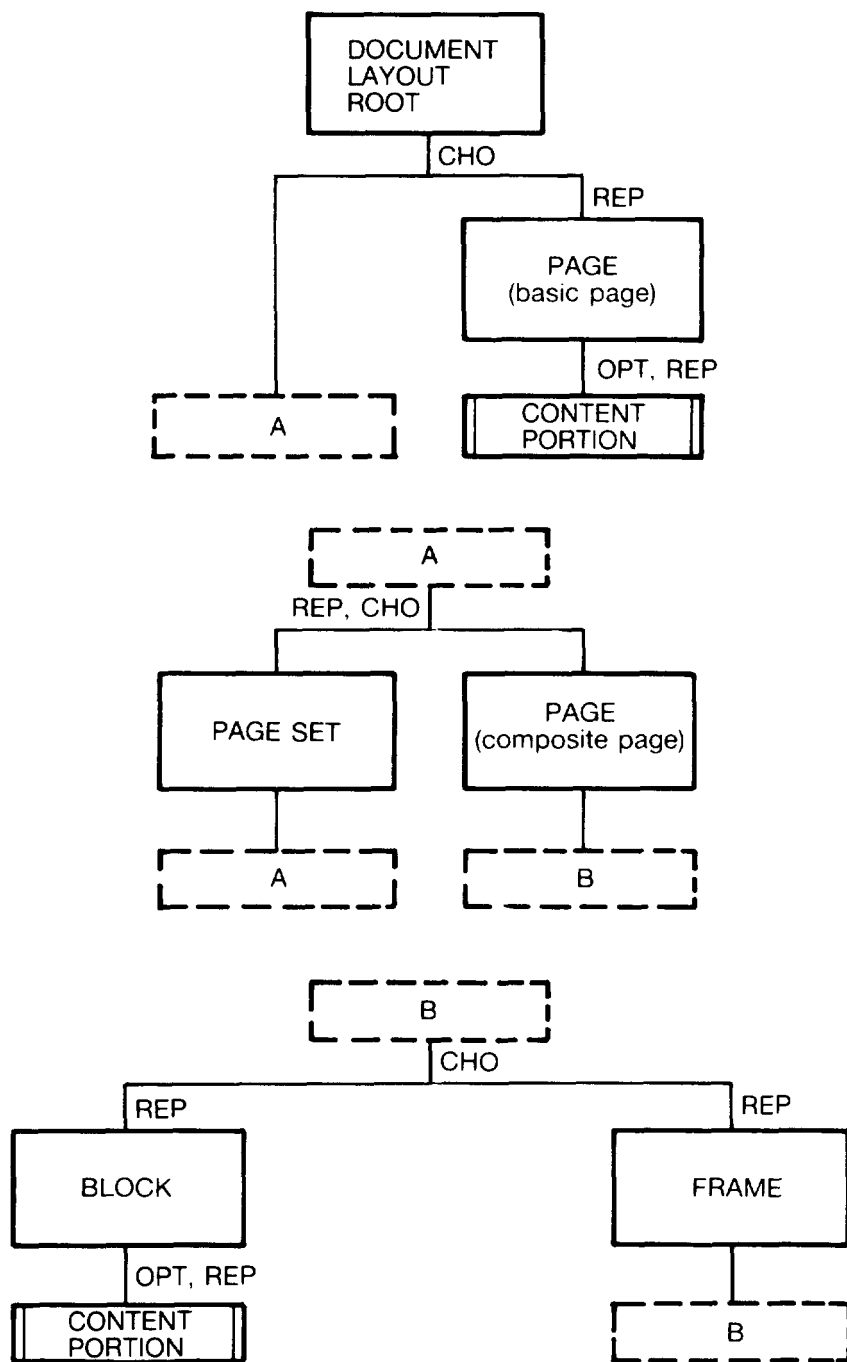


Figure 6 — Permissible specific layout structures

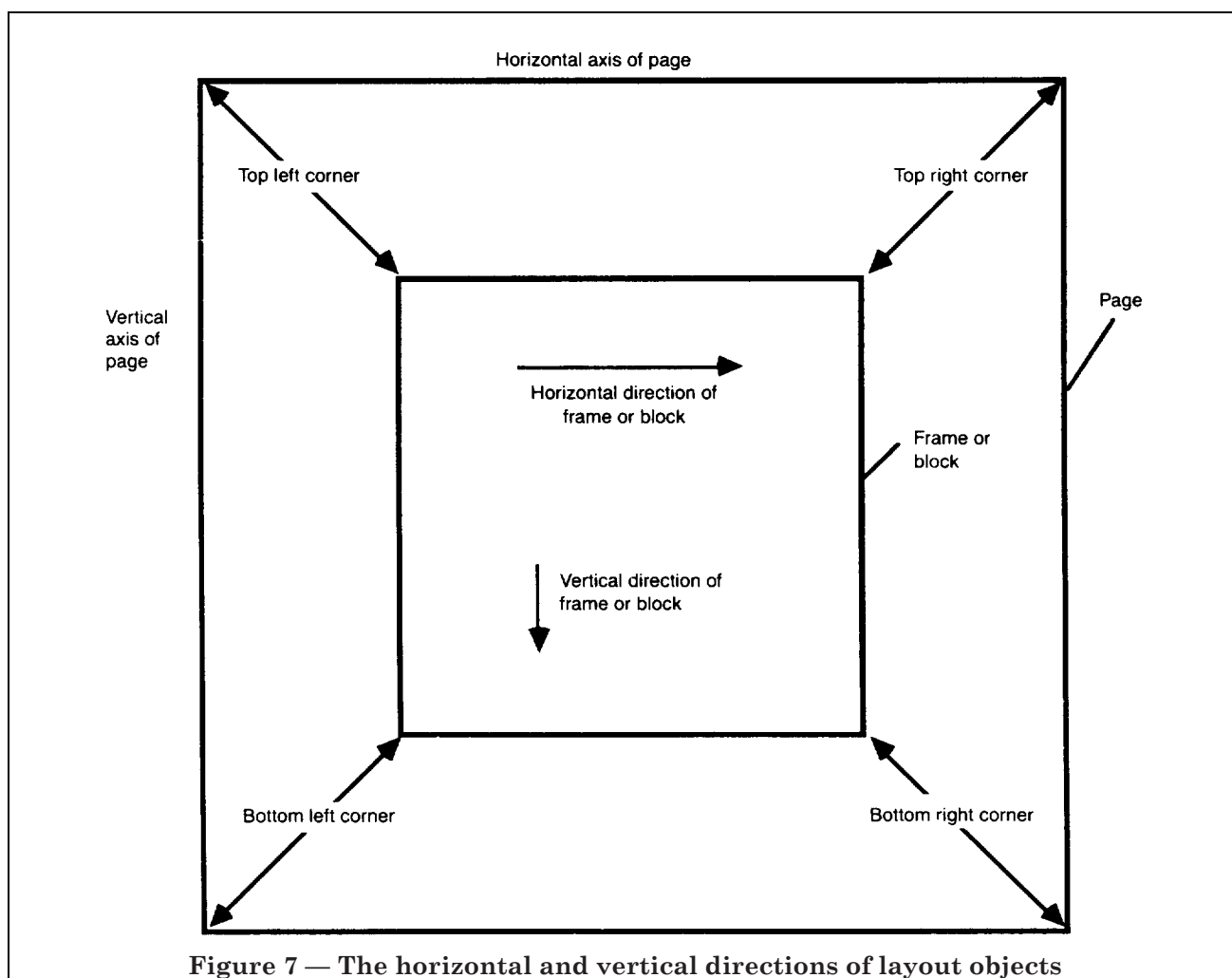


Figure 7 — The horizontal and vertical directions of layout objects

3.3.2.2 Positioning of frames

The reference point for positioning a frame is the top left corner of that frame. Frames are positioned relative to the reference point of the object to which they are immediately subordinate. Thus, frames that are immediately subordinate to a page are positioned relative to the origin of the page co-ordinate system. Frames which are immediately subordinate to another frame are positioned relative to the reference point of that frame.

3.3.2.3 Positioning of blocks

The reference point for positioning a block is the top left corner of that block. Blocks are positioned relative to the reference point of the layout object to which they are immediately subordinate.

Thus, blocks that are immediately subordinate to a page are positioned relative to the origin of the page co-ordinate system. Blocks that are immediately subordinate to a frame are positioned relative to the reference point of that frame.

3.3.3 Naming of edges of frames and blocks

In the following definition the layout path (see 5.4.2.2, 6.5 and 6.6) referred to is that specified for a frame, or in the case of a block for its immediately superior frame.

For a frame or block, the leading and trailing edges are defined as the two opposite edges of the same frame or block that are orthogonal to the direction of layout path, such that the direction from the trailing edge to the leading edge is in the same direction as the layout path.

The left-hand and right-hand edges of a frame or block are defined as the two opposite edges of the same frame or block that are parallel to the direction of layout path, such that the direction from the right-hand edge to the left-hand edge is at an angle of 90° anti-clockwise relative to the direction of the layout path.

The names of the edges of frames and blocks are illustrated in Figure 8.

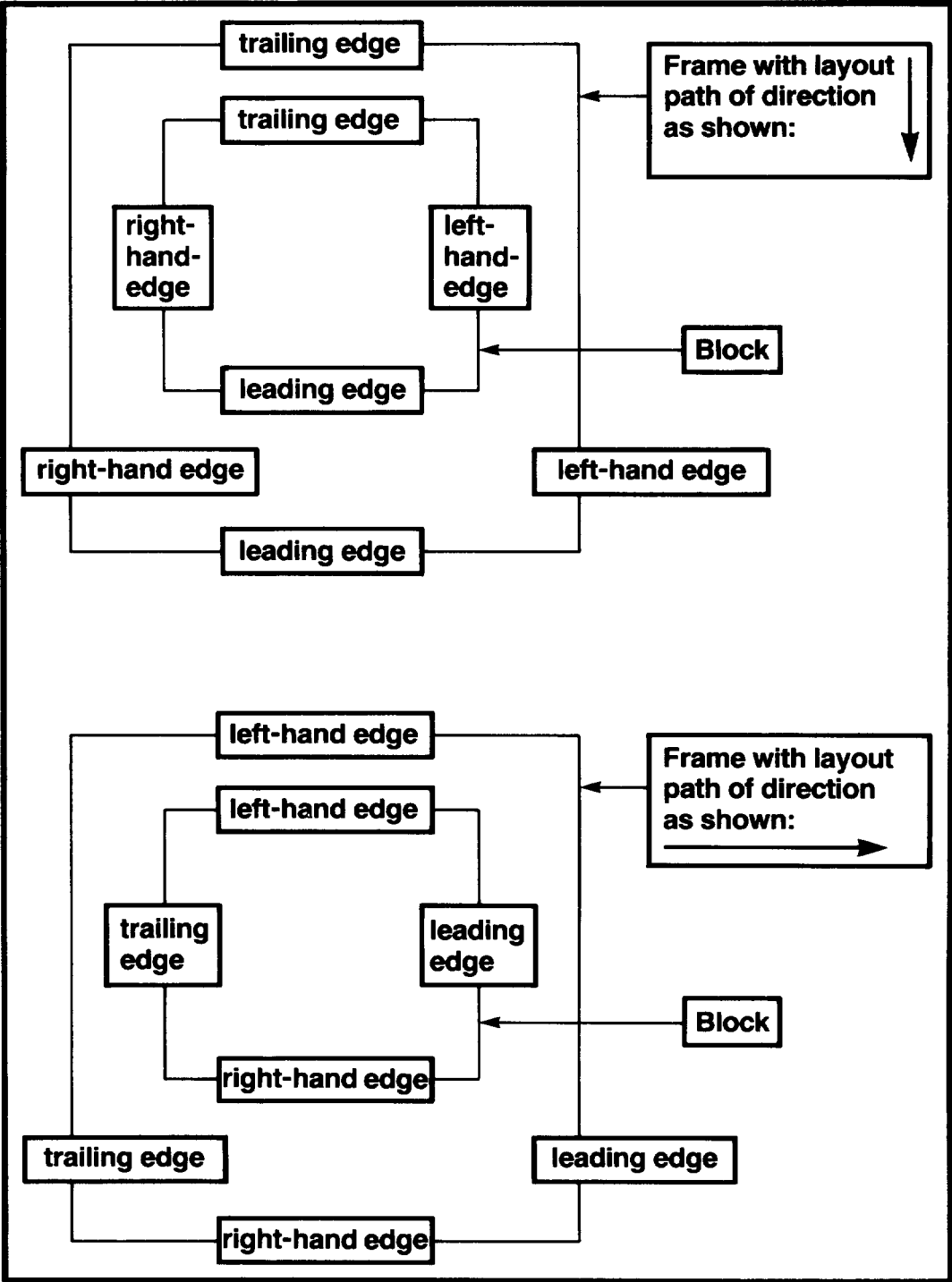


Figure 8 — Naming of the edges of blocks and of their immediately superior frames

3.3.4 Measurement

3.3.4.1 Basic measurement unit

For the purpose of conveying the originator's intentions, all dimensions and positions are expressed in basic measurement units (BMU). The value of the *basic measurement unit* is equal to 1/1 200 of 25,4 mm. A locally defined scaling factor may be used to map the document to a particular imaging device.

3.3.4.2 Scaled measurement units

All attributes and numeric control function parameters that specify absolute or relative positions and dimensions are expressed as integral multiples of a *scaled measurement unit (SMU)* that is equal to m/n basic measurement units. The *unit scaling* factor is specified in the form of two integers m and n by the document profile attribute "unit scaling" (see ISO 8613-4). When the attribute "unit scaling" is not specified, the scaled measurement unit used in the positioning and dimensioning attributes is equal to the basic measurement unit.

3.3.5 Borders of frames and blocks

Borders can be specified to be around the edges of blocks or within the edges of frames.

A *border* consists of the border freespace and the border line (see Figure 9 and Figure 10).

The border line is described by specifying its width and line type. Line type can take the values solid, dashed, dot, dash-dot, dash-dot-dot or invisible.

The border freespace is described by specifying its width.

The border may be specified for a particular set of edges of the frame or block, or for all edges. The characteristics of the border for each of the edges may have a separate specification.

3.3.5.1 Borders of blocks

A border around a block is illustrated by Figure 9.

In the case of a block, the border freespace provides a region which surrounds the block between the block boundary and the border line.

The border surrounds the block and does not intrude on the block. The border is contained entirely within the area of the object to which the block is immediately subordinate. This means that the document layout process must take the border into account in determination of the available area.

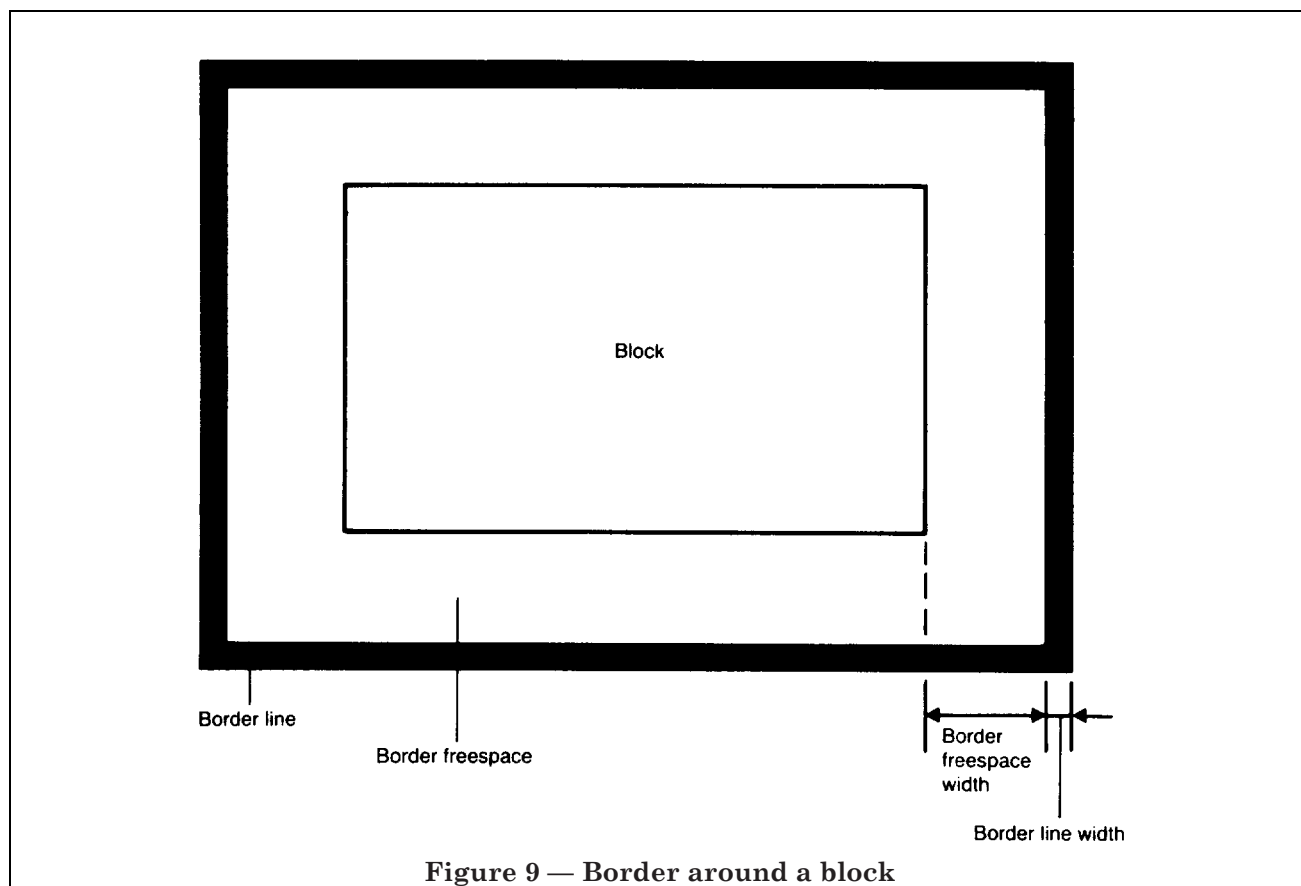
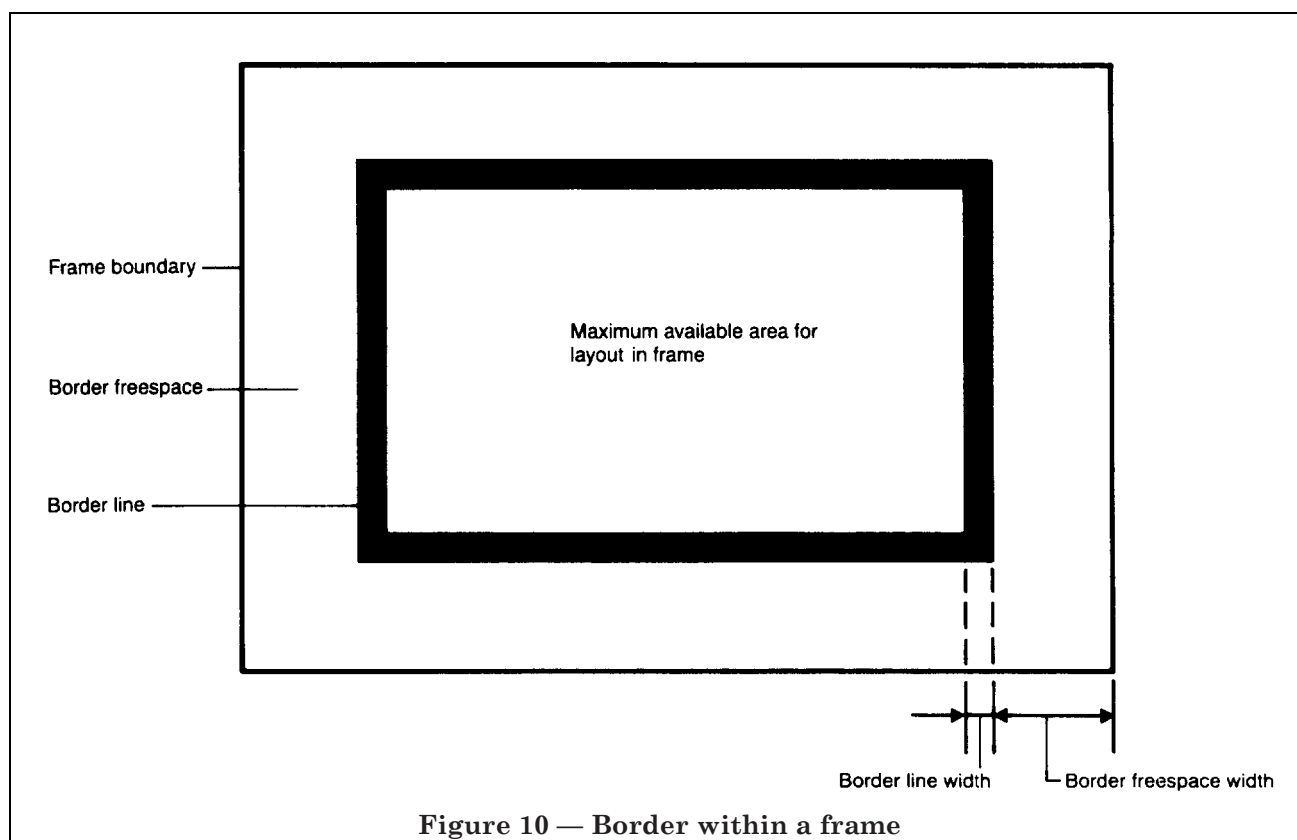


Figure 9 — Border around a block



3.3.5.2 Borders of frames

A border within a frame is illustrated by Figure 10.

In the case of a frame, the border freespace provides a region which is within the frame between the frame boundary and the border line.

The border reduces the available area within the frame for layout purposes. This means that the document layout process must take the border into account in determination of the available area.

3.4 Documents containing both specific structures

When a document contains both a specific layout structure and a specific logical structure, each content portion in the specific structure is in general associated with both of the structures. However, some of these content portions can be associated with the specific layout structure only. These are the content portions that

- represent formatted content corresponding to generic content portions associated with a basic logical object class of the generic logical structure or resource-document;
- are created as a result of the attribute “content generator”;

— are created as a result of the attribute “logical source” of a layout object class description.

The allocation of content portions to basic objects in the specific layout structure may not correspond to the allocation of content portions to basic objects in the specific logical structure.

For example, a basic logical object representing a paragraph would need to contain at least two content portions in the case where the paragraph was split across a page boundary. This is shown in Figure 11 if the middle basic logical object represents the split paragraph and the composite layout objects represent the two separate pages.

This is in contrast to the situation when a document contains only one specific structure, in which case there is no need for dividing the content associated with a basic component into more than one content portion.

Figure 11 also illustrates the correspondence between logical and layout objects in the case that there is both a specific logical structure and a specific layout structure.

The presentation attributes for a content portion are specified by the layout structure and referenced presentation styles, any specified by presentation styles referenced by the logical structure are ignored.

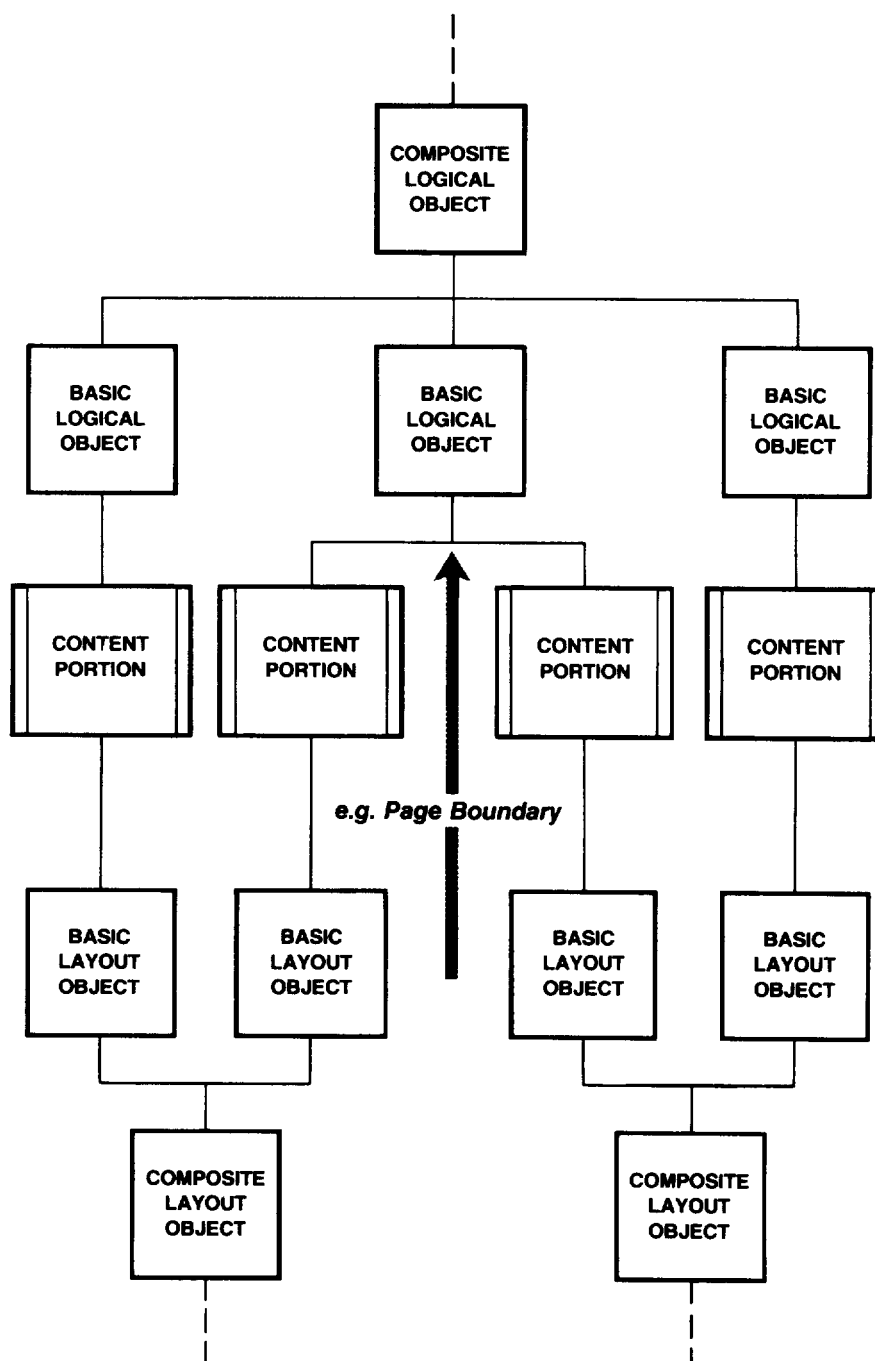


Figure 11 — Example of relationship between logical and layout objects and associated content portions

In the imaging process, the logical structure and referenced styles are ignored.

3.5 Generic structures

3.5.1 General principles

The generic structure of a document describes characteristics common to a number of objects within a document. In the most comprehensive case it describes the common characteristics of a document class.

The generic structure can be used to

- improve transmission efficiency by factorisation;
- maintain the internal consistency of a document by providing the recipient with the structural information necessary to edit and/or lay out the document as intended by the originator.
- facilitate the creation of objects and documents by the recipient as prepared by the originator.

The generic structure consists of a generic logical structure and/or a generic layout structure.

Within the set of constituents representing the generic structure each object class description consists of attributes which parallel the attributes of object descriptions. These attributes can be used to determine the value(s) for the attributes of the corresponding object descriptions. The object class descriptions may also contain references to layout and presentation styles.

If an attribute value is specified explicitly in an object description then this overrides any value that may be derived from the corresponding attribute of an object class description.

Each object class in a generic structure specifies the object type of all the objects that are derived from the object class. There may be many object classes of the same object type. The object types that can occur are specified in 3.5.6 and 3.5.7.

Each object class in a generic structure is uniquely identified within that structure.

Object classes for basic objects may specify content, either in the form of generic content portions or by specifying an expression to generate content.

3.5.2 Generator and factor sets of object class descriptions

A complete generator set of logical object class descriptions can be used to control the editing process. In a document with such a complete generator set every logical object description references a logical object class description.

In addition, the generation of immediate subordinates is completely derived from the referenced logical object class descriptions.

A partial generator set of logical object class descriptions can be used to guide the generation of sub-hierarchies of the specific logical structure during the editing process. In a partial generator set, some composite logical object class descriptions may constrain the possible subordinates for objects of the class, while others do not.

A factor set of logical object class descriptions can be used to guide the creation of logical objects during the editing process. In a factor set, the composite logical object class descriptions do not constrain the possible subordinates for objects of the class. Such a factor set need not include an object class description corresponding to the document logical root.

A complete generator set of layout object class descriptions can be used to control the layout process. In a document with such a complete generator set every reference from the layout directive attributes, either by layout reference or layout category, is satisfied in the layout object class descriptions.

If the constituents representing the specific layout structure are present together with such a complete generator set then every composite layout object description references a layout object class description; and the generation of immediate subordinates is completely derived from the referenced layout object class descriptions. A block may or may not reference a layout object class description.

A partial generator set of layout object class descriptions can be used to guide the generation of sub-hierarchies of the specific layout structure. In a partial generator set, some composite layout object class descriptions may constrain the possible subordinates for objects of the class, while others do not.

A factor set of layout object class descriptions can be used to guide the creation of layout objects. In a factor set, the composite layout object class descriptions do not constrain the possible subordinates for objects of the class. Such a factor set need not include an object class description corresponding to the document layout root.

In any of the preceding cases, object class descriptions achieve factorisation by holding information that is common to a number of object descriptions. Such factorisation of attribute values, by avoiding replication, can result in improved transmission efficiency and can also facilitate the creation of objects by the recipient. Such factorisation applies to content when an object class description either specifies generic content portion description(s) or specifies an expression to generate content.

3.5.3 Complete generator sets of object class descriptions

A complete generator set of layout object class descriptions form the nodes of a single directed graph.

A complete generator set of logical object class descriptions form the nodes of a set of directed graphs. This set consists of a primary graph and, optionally, one or more secondary graphs each corresponding to the logical object class descriptions referenced by the attribute “logical source” (see 5.4.2.3).

Each directed arc of each graph connects the node from which it starts with each of the nodes which represent a possible class of immediately subordinate objects that can be generated. For any node each possible class of immediately subordinate objects is represented by a directed arc starting from the node and ending on a node representing the class of the immediately subordinate objects.

The rules describing the possible hierarchical relationships between a composite object and its immediate subordinates, which correspond to these arcs, are defined in the object class description of the composite object.

Each graph is connected such that there is a single root node which is such that

- it is possible to reach every other node on a path through the graph following the directed arcs;
- there is no path through the graph following the directed arcs from which it is possible to reach this single root node.

The single root node of the graph represents the object class description of object type document layout root in the set of constituents representing the generic layout structure.

The single root node of the primary graph represents the object class description of object type document logical root in the set of constituents representing the generic logical structure.

The single root node of each secondary graph represents an object class description which is identified by the attribute “logical source” of one or more layout object class descriptions.

Some paths may pass through a particular node a multiple number of times. Some of the directed arcs may start and end on the same node.

The non-terminal nodes, that is the nodes from which some directed arc starts, which in all cases include the document root itself, represent object classes for composite objects.

In the case of the terminal nodes, that is the nodes from which no directed arc starts, the situation depends on which generic structure is represented by the complete generator set of object class descriptions.

If the generic logical structure is represented, then the terminal nodes represent the object classes for basic logical objects.

If the generic layout structure is represented, then the terminal nodes represent object classes for basic pages, for blocks or for frames which will always be the lowest level frame in the particular branch of the hierarchy (see 3.5.7).

Generic content portions are only referenced from object classes for basic objects and each generic content portion must be referenced by just one object class.

3.5.4 Generation of structures

A complete generator set of object class descriptions and associated generic content portions representing a generic structure is used to maintain the internal consistency of a document. This is achieved by defining which specific structures are possible in a given document class and which object classes are possible in parts of a document (see 5.3.2.1). When creating, editing or laying out a document, a generic structure can be used to control the generation and modification of the specific structure and thereby preserve the intentions of the originator, in this case the creator of the document class description.

Generation and modification of the specific structure is controlled by ensuring that the attributes of each object class description are used to specify default values for attributes in the corresponding object descriptions.

Recursion is permitted in the generic logical structure and in the generic layout structure, thus objects of the same object class may be hierarchically related.

ISO 8613 does not define any particular generic structures or object classes. However, it allows the definition of a potentially infinite variety of document classes.

3.5.5 Content in generic structures

An object class description for a basic object can specify content in one of three ways:

- by specification of one or more generic content portions;
- by specification of a content generator;
- by referencing an object class in a resource-document which has content specified.

A content generator allows the content information to be specified by an expression which is evaluated during the layout process. Together with specification of a content generator a content portion description may also be present, in order to allow for specification of attributes of the content portion.

When an object class description specifies content, content portion descriptions can be derived for the specific structure.

A basic object class description with a generic content portion description may be referred to by more than one basic object description of the same specific structure and of the same object class, which permits content sharing.

3.5.6 Generic logical structure

The object classes that can occur within the generic logical structure of a document are for objects of the following object types:

- document logical root;
- composite logical object;
- basic logical object.

The possible hierarchical relationships between logical object classes are as follows:

- the object class for the document logical root can specify that there shall be any number and combination of immediately subordinate composite logical objects and basic logical objects;
- an object class for a composite logical object can specify that there shall be any number and combination of immediately subordinate composite logical objects and basic logical objects;
- an object class for a basic logical object can optionally have content specified; if content is specified then it may be in the form of one or more generic content portions, a reference to an object class in a resource-document which has content specified, or a content generator.

Object classes of these object types serve as templates for the corresponding objects in the specific logical structure.

3.5.7 Generic layout structure

The object classes that can occur within the generic layout structure of a document are for objects of the following object types:

- document layout root;
- page set;
- composite or basic page;
- frame;
- block.

The possible hierarchical relationships between layout object classes are as follows:

- the object class for the document layout root can specify that there shall be any number and combination of immediately subordinate pages or page sets;
- an object class for a page set can specify that there shall be any number and combination of immediately subordinate pages or page sets;
- an object class for a composite page can specify that there shall be either any number of immediately subordinate frames or any number of immediately subordinate blocks;
- an object class for a frame can specify that there shall be either any number of immediately subordinate frames or any number of immediately subordinate blocks;
- an object class for a basic page or a block always has content specified; this may be by specifying one or more generic content portions, by specifying a content generator, or by referencing an object class in a resource document which has content specified.

Object classes of these object types serve as templates for the corresponding objects in the specific layout structure. Basic objects created by the layout process, for the content associated with the logical structures, do not reference layout object classes.

3.5.8 Resource-document

A logical object class description may contain a reference to a logical object class description in the resource-document (see 2.3.10). A layout object class description may contain a reference to a layout object class description in the resource-document. In both cases, a subset or all of the attributes may be present in the object class description of the interchanged document only implicitly, by reference to the object class description of the resource-document. This may include the attribute “content portions”, in which case content portions may be present in the interchanged document only implicitly, by reference to the content portion descriptions of the resource-document.

The association between the interchanged document and the resource-document is achieved through the use of resource names. The resource-document includes, within the document profile, a table which maps resource names onto the object class identifiers of that resource-document. An object class description within the interchanged document may then refer to an object class description within the resource-document by using one of these resource names.

Attributes in the resource-document that are used by reference as attributes of object class descriptions in the interchanged document shall not make use of object class identifiers.

Section 4. Interface between the document architecture and content architectures

4.1 General concept

A document is defined by means

- of the document architecture: a finite set of constituents together with their characteristics and rules that establish their relationships;
- of content architectures: a finite set of content elements, (for example, graphic characters, pels), together with their characteristics and rules that establish their relationships.

Content elements and control functions make up content portions governed by a content architecture. Every content portion is associated with a basic component governed by the document architecture. One content element is thus specified within the document through a specific content architecture together with the document architecture.

The terms basic values and non-basic values are used in this section

- *basic values* of attributes, control function parameters and other capabilities are those that are unconditionally allowed in document interchange in the context of a particular document application profile;
- *non-basic values* of attributes, control function parameters and other capabilities are only allowed in document interchange in the context of a particular document application profile, if their use is declared in the document profile.

The document architecture and any content architectures are connected through an interface, as shown in Figure 12.

The interface is specified by describing which attributes defined in a content architecture have influence on constituents of the document architecture (document profile, basic component descriptions, content portions).

4.2 Specification of a content architecture

The specification of a content architecture consists of three categories of information:

- structure information that identifies the class of content architecture, its internal structure rules and the positioning and imaging rules;
- content information that determines what information comprises the content portion. It includes a set of content element repertoires along with a default repertoire and the set of control functions available in the content architecture;

- information that specifies the basic values, default values and non-basic values of presentation attributes, coding attributes and control function parameters.

This information is required in order to link the document architecture to content architecture(s), through the interface information.

4.3 Interface information

A content architecture has the following three interfaces to the document architecture:

- the document profile, which includes attributes that identify the content architecture class(es) used within the document and attributes that specify the use of any non-basic content architecture features;
- an object description or an object class description for a basic object, which include attributes that identify the content architecture class and presentation attributes;
- a content portion description, which includes attributes that identify the type of coding and coding attributes.

The definition of each content architecture specifies the information associated with these attributes as described in the following sub-clauses.

4.3.1 Interface between a content architecture and the document profile

The definition of a content architecture shall include the following information:

- the value(s) of the attribute used to identify the content architecture class in the document profile;
- the format of the attribute used to specify non-basic values of presentation attributes in the document profile;
- the format of the attribute used to specify default values of presentation attributes in the document profile;
- the format of the attribute used to specify non-basic values of coding attributes in the document profile;
- the format of the attribute used to specify default values of coding attributes in the document profile.

The distinction between basic and non-basic values of presentation attributes and coding attributes is not specified as a part of the definition of a content architecture. This distinction is made in the definition of a document application profile in accordance with ISO 8613-1.

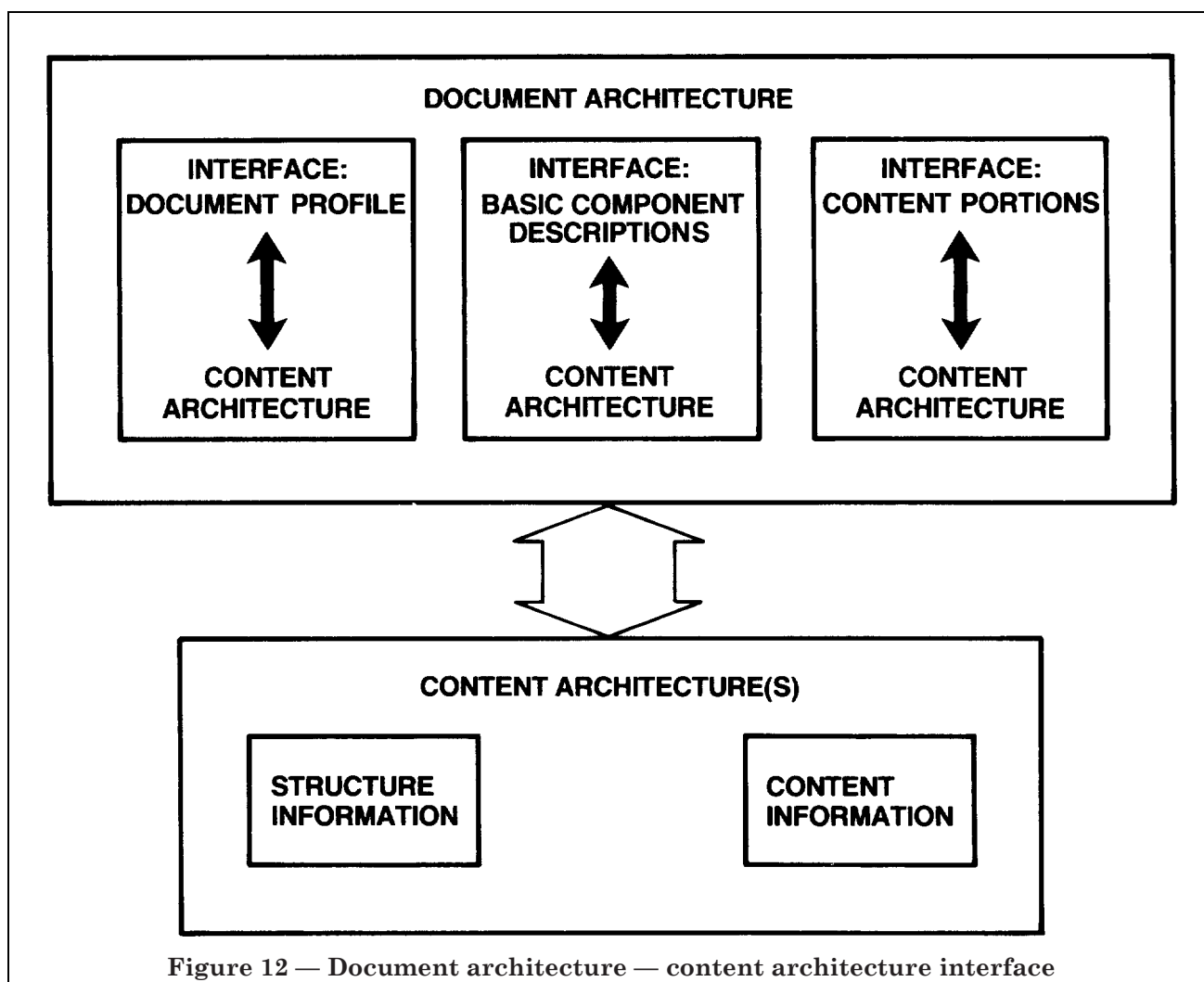


Figure 12 — Document architecture — content architecture interface

4.3.2 Interface between a content architecture and a basic component description

A definition of a content architecture shall include the following information:

- the value(s) of the attribute used to identify the content architecture class in a basic component description;
- the formats, permissible values and recommended default values of the presentation attributes;
- the presentation attributes for which a document application profile may define non-standard default values;
- possible interactions between presentation attributes and document architecture attributes.

4.3.3 Interface between a content architecture and a content portion

The definition of a content architecture shall include the following information:

- the value(s) of the attribute used to identify the type of coding in a content portion;
- the formats, permissible values and recommended default values of the coding attributes;
- the coding attributes for which a document application profile may define non-standard default values;
- possible interactions between control functions and presentation attributes.

Section 5. Attribute definitions

5.1 General principles of attributes

Characteristics of structural elements of a document and relationships between structural elements are represented by constituents which are sets of attributes. Each attribute is identified by a name and has a value that describes the characteristic or relationship. Attributes are also used to identify constituents.

This section contains definitions of all attributes defined in the document architecture. The attribute definitions specify the range of values that each attribute can assume.

5.1.1 Attribute categorisation

Attributes are categorised according to the constituent to which they apply, as follows:

- document profile attributes;
- component description attributes;
- layout style attributes;
- presentation style attributes;
- content portion description attributes.

The constituents termed descriptions are those which are counterparts of the structural elements (objects, object classes and content portions).

Document profile attributes are defined in ISO 8613-4. The general principles of other attribute categories are described in this clause and the attributes are defined in 5.3 to 5.9.

NOTE The tables in Annex F summarise all attributes defined in this section, in accordance with their categorisation.

5.1.1.1 Attributes of components

Attributes of component descriptions are further categorised as follows:

- shared attributes, these can be included in both logical and layout component descriptions (see 5.3);
- layout attributes, these can be included in layout component descriptions only (see 5.4);
- logical attributes, these can be included in logical component descriptions only (see 5.5);

Some attributes of components can be included in composite component descriptions only, some in basic component descriptions only and some only in component descriptions of a particular object type.

In addition, some attributes may be included in object class descriptions only, some in object descriptions only and some in both object class descriptions and in object descriptions.

All attributes, except the presentation attributes (see 5.1.1.4), are independent of the content architecture pertaining to any component descriptions.

5.1.1.2 Layout style attributes

A layout style consists of

- a layout style identifier;
- a user-visible name;
- user-readable comments; and,
- a set of layout directive attributes.

Layout styles are referred to from logical component descriptions. Such a reference is made using the attribute “layout style” with a value equal to the value of the layout style identifier. The layout style identifier uniquely identifies the layout style within the document.

Layout styles are defined in 5.6.

A layout style may be referenced by more than one logical component description. The effect of a reference to a layout style is to apply its layout directives to that logical component description which contains the reference. Precedence rules are specified in 5.1.2.4 and 5.7.12.

The layout directives are a set of attributes which specify information for the document layout process (see section 6). The layout directives guide the document layout process in generating the specific layout structure.

Layout directive attributes are defined in 5.7.

Thus, layout styles affect the layout of objects, not their content and are independent of particular content architectures. Presentation styles affect the layout and imaging of the content associated with basic objects and hence are content architecture specific. Therefore, there is no conflict between the two.

5.1.1.3 Presentation style attributes

A presentation style consists of

- a presentation style identifier;
- a user visible name;
- user-readable comments;
- transparency;
- colour;
- border; and,
- sets of presentation attributes.

A presentation style may be referred to from a basic logical or basic layout component description. Such a reference is made using the attribute “presentation style” with a value equal to the value of the presentation style identifier. The presentation style identifier uniquely identifies the presentation style within the document.

A presentation style may be referenced by more than one component description.

Presentation styles are defined in 5.8.

The effect of a reference to a presentation style is to apply its presentation attributes, transparency, colour and border to the basic component which contains the reference. In addition, presentation attributes can be specified for basic layout components. Precedence rules are specified in **5.1.2.4**.

Each set of presentation attributes corresponds to a particular content architecture class.

Presentation styles affect the layout and imaging of the content associated with basic objects and hence are content architecture specific. Layout styles affect the layout of objects, not their content. Therefore, there is no conflict between the two.

5.1.1.4 Presentation attributes

Presentation attributes, which can be applied to basic components only, are used to specify the properties of the content portion(s) associated with that component. The presentation attributes specify information for the content layout process and content imaging process.

The presentation attributes specify the initial conditions concerned with the presentation of the content portion(s). The presentation attributes are specified either in a presentation style or, for layout components, in a basic layout component description.

The set of presentation attributes that is applicable depends upon the particular content architecture class specified for the basic component. The content architecture class attributes specify the content architecture class of the associated content portion(s), and therefore which set of presentation attributes are applicable to the content (see **5.3.4**).

The presentation attributes are defined in those parts of ISO 8613 that are concerned with individual content architectures.

5.1.1.5 Content portion description attributes

A content portion description consists of

- content identifier-logical;
- content identifier-layout;
- type of coding;
- alternative representation;
- coding attributes; and,
- content information.

These attributes uniquely identify the content portion description, specify the type of coding used to code the content information and specify an alternative representation that may be imaged in lieu of content information when a recipient is not capable of decoding and/or imaging the content portion.

Content portion identifiers and alternative representation are fully specified in **5.9**. Other attributes of content portions depend on the content architecture and details are specified in those parts of ISO 8613 that deal with individual content architectures.

Content portion descriptions are defined in **5.8**.

5.1.2 Attribute values

5.1.2.1 General principles

Each constituent consists of a set of attributes, the attributes in the set are said to be *specified for* the particular constituent.

In the case of object descriptions, the complete set of attributes describing the characteristics of the object need not be specified for the constituent, such attributes can also be derived from attributes specified for other constituents, by using defaulting rules specified in **5.1.2.4**. In such cases, the attributes which are specified for the constituent, together with those derived for the constituent, are said to form the set of attributes which *apply to* the constituent (or that are applied to, or are applicable to, the constituent).

Attributes may be structured into parameters and sub-parameters, each attribute definition includes a specification of any such structuring.

The definition of an attribute also indicates whether the value(s) of that attribute may consist of one or more elements of data or expression(s).

Attribute values in the document profile, presentation styles and content portion descriptions can only consist of data elements of defined types; attribute values in component descriptions and layout styles may consist of either data elements or expressions.

If the value of an attribute consists of one or more data elements, then each element will be of a certain type and will be taken from a predefined set of values.

Such a value may consist of

- a) a sequence of numeric values;
- b) a string of characters from a defined character set;
- c) an element from a set of data elements defined for that attribute;
- d) a reference to another constituent that exists within the document;
- e) a reference to a value of an attribute of a constituent that exists in the document.

An example of an attribute value of type c) is the sub-parameter “fill order” of the attribute “position” (see 5.4.1.1) which can have a value equal to a member of the set { “normal order”, “reverse order”}. An example of an attribute value of type d) is the attribute “presentation style”. An example of an attribute value of type e) is the parameter “binding name” of the attribute “bindings”.

Alternatively, if the value of an attribute consists of an expression, then the expression will need to be evaluated in order for the value to be interpreted. This value may depend upon the values of attributes in other constituents (see 5.1.3).

5.1.2.2 Attribute classification

Attributes can be classified as mandatory, defaultable or non-mandatory.

The classifiers are m, d, nm, as follows:

- m — Mandatory attribute: The attribute shall be specified for the constituent.
- d — Defaultable attribute: The attribute need not be specified for the constituent; the value can be derived using the defaulting mechanism specified in 5.1.2.4
- nm — Non-mandatory attribute: The attribute need not be specified for the constituent. If the attribute is not specified for a constituent, the attribute does not apply to that constituent.

Attributes of object descriptions, content portion descriptions and styles when applied to object descriptions, can be classified as either mandatory, defaultable or non-mandatory; attributes of all other constituents can only be classified as mandatory or non-mandatory.

In the case of defaultable attributes, the attribute definition also specifies the standard default value (see 5.1.2.4, 5.1.2.5).

A document application profile may specify non-standard default values for attributes (see ISO 8613-4). If this is the case they are declared in the document profile by means of the document profile attribute “document application profile defaults”.

5.1.2.3 Default value lists

Certain attributes that are classified as defaultable may be specified in a default value list. Default value lists can be specified for composite component descriptions.

The purpose of a default value list is to allow the values of some attributes, which describe characteristics of objects in the specific layout and specific logical structures, to be specified in constituents that correspond to objects at higher levels of the hierarchical structure. The use of a default value list forms part of the defaulting mechanism defined in 5.1.2.4.

A composite component description can specify one or more default value lists. If more than one list is specified, each list applies to a different subordinate object type. Each default value list applies to all subordinate objects of the relevant object type.

For example, a default value list specified for a page can apply to subordinate frames or to subordinate blocks within the page.

5.1.2.4 Determining values for defaultable attributes of objects

The values of defaultable attributes of objects can be derived from

- 1) within the object description;
- 2) within the object class description;
- 3) within a style;
- 4) within a default value list at higher levels of the hierarchical structure;
- 5) within an object class description in a referenced resource-document;
- 6) within a list of default values in the document profile, representing default values specified by a document application profile;
- 7) within ISO 8613.

In case 4) the attributes are interpreted as default values for the lower levels. They can be overridden by attributes of object descriptions, styles or object class descriptions specified at the lower levels.

For example, it is possible to specify

- at document layout root level, the default page size;
- at page level, the default line spacing for blocks containing character content.

The default values for attributes applicable to logical object descriptions are determined in the sets of constituents representing the specific and generic logical structures and referenced presentation and layout styles.

The default values for attributes applicable to layout object descriptions are determined in the sets of constituents representing the specific and generic layout structures and referenced presentation styles.

To determine the value of an attribute of an object that is classified as defaultable, the value is determined by the first of the following rules which is applicable.

When an attribute is structured into two or more parameters then it can be specified that the defaulting rules are to apply to each parameter independently. When this is possible it is explicitly specified in the attribute definition, in the part of the definition which specifies the default values.

- a) If an attribute value is specified for the object description concerned, then that value is used.
- b) If the object description concerned refers to a style, and that style contains a value for the attribute, then that value is used.
- c) If the object description concerned refers to an object class description, and that object class description contains a value for the corresponding attribute, then the value of the attribute is derived from that corresponding attribute.
- d) If the object description concerned refers to an object class description, which contains a reference to a style which in turn contains a value for the attribute, then that value is used.
- e) If the object description concerned refers to an object class description which refers to an object class description in the resource-document, and the object class description in the resource-document contains a value for the corresponding attribute, then the value of the attribute is derived from that corresponding attribute.
- f) If the object description concerned refers to an object class description which refers to an object class description in the resource-document which contains a reference to a style which in turn contains a value for the attribute, then that value is used.
- g) If a default value list specified at a higher level of the hierarchical structure is applicable to the object (that is, it is specified for this object type), and that default value list contains a value for the attribute, then the value of the attribute is derived from the value specified in that default value list.

If more than one such default value list specifies a value for the same attribute, then the value derived from the lowest hierarchical level in the structure is used.

At each level, if a default value list exists for both an object description and an object class description, then only default values which are not specified in the default value list of the object description are taken from the default value list of the object class description.

At each level, if the default value list exists for an object class description in the resource-document that is referred to by an object class description that is referred to by an object description, then only default values which are not specified in the object description or in the object class description in the interchanged document are taken from the object class description in the resource-document.

At each level the default value list may specify an attribute value directly, or indirectly via a style. In the case that an attribute value is specified in both ways in a single default value list, the value specified explicitly in the default value list is used.

- h) If no value is determined by the preceding steps a) to g), and a default value is defined for this attribute by the document profile attribute “document application profile defaults” (see ISO 8613-4) then that value is used
- j) If no value is determined by the preceding steps a) to h), then the default value defined in ISO 8613 is used.

5.1.2.5 Determining values of attributes of content portions

The content associated to a basic object in a specific structure is determined by the first of the following to specify either any content portions or a content generator:

- the basic object description;
- an object class description referenced from the basic object description;
- an object class description in the resource-document referenced from an object class description referenced from the basic object description.

To determine the value of an attribute of a content portion that is classified as defaultable, the value is determined by the first of the following rules which is applicable.

- a) If an attribute value is specified for the content portion description concerned, then that value is used;
- b) If no value is determined by step a), and a default value is defined for this attribute by the document profile attribute “document application profile defaults” (see ISO 8613-4) then that value is used;

- c) If no value is determined by the preceding steps a), b), then the default value defined in ISO 8613 is used.

5.1.3 Expressions

The value of some attributes can be specified by an expression. These attributes are

- “generator for subordinates”;
- “content generator”;
- “bindings”, for the parameter “binding value”;
- “same layout object”, for the first parameter;
- “synchronisation”.

The expression permitted in the attribute “generator for subordinates” is defined in 5.3.2.1. The other possible expressions are defined in this sub-clause.

There are three types of expressions:

- a) string expressions (see 5.1.3.1);
- b) numeric expressions (see 5.1.3.2);
- c) object identifier expressions (see 5.1.3.3).

A string expression or a numeric expression may refer to a binding value (see 5.1.3.4).

5.1.3.1 String expressions

A string expression within an attribute value specification consists of either an atomic string expression or a sequence of two or more atomic string expressions.

An atomic string expression is one of the following:

- a) a string literal;
- b) a reference to a binding value (see 5.1.3.4);
- c) a string function application (see below).

A string literal is an arbitrary octet string.

NOTE This octet string may be interpreted in various ways, depending on the application. For example, as a character string of a particular character repertoire, as a facsimile image or as a geometric picture.

A string function application is an application of one of the functions:

- **MAKE-STRING**: this function produces a character string consisting of the decimal numeral representing the numeric value of the parameter, which may be any integer (negative, zero or positive integers are permitted);
- **UPPER-ALPHA**: this function produces a character string consisting of one of the upper case letters A to Z, corresponding to the numeric value (1 to 26) of the parameter;
- **LOWER-ALPHA**: this function produces a character string consisting of one of the lower case letters a to z, corresponding to the numeric value (1 to 26) of the parameter;

- **UPPER-ROMAN**: this function produces a character string consisting of the Roman numeral representing the numeric value of the parameter, which may be any positive integer, using the upper case letters: C, D, I, L, M, V and X;

- **LOWER-ROMAN**: this function produces a character string consisting of the Roman numeral representing the numeric value of the parameter, which may be any positive integer, using the lower case letters: c, d, i, l, m, v and x.

Each of these functions has a parameter consisting of a numeric expression (see 5.1.3.2).

A string function application produces an empty string when the value of the parameter is out of range.

When two or more atomic string expressions occur in sequence each expression is evaluated independently and their results are concatenated.

5.1.3.2 Numeric expressions

A numeric expression within an attribute value specification consists of one of the following:

- a) a numeric literal;
- b) a reference to a binding value (see 5.1.3.4);
- c) a numeric function application (see below).

A numeric literal is any integer (i.e. negative, zero or positive integers are permitted).

A numeric function application is an application of one of the functions:

- **INCREMENT** — this function has one parameter, consisting of a numeric expression. The result of the function is a numeric value which is one greater than the value of the parameter;
- **DECREMENT** — this function has one parameter, consisting of a numeric expression. The result of the function is a numeric value which is one less than the value of the parameter;
- **ORDINAL** — this function has one parameter, consisting of a reference to an object, represented by either an object identifier or an object identifier expression.

The result of the function is equal to the sequence number of the specified object, within the set of objects that are immediately subordinate to the immediate superior of the specified object and belong to the same object class as the specified object. Within the set of objects that satisfy this criterion, the objects are ordered according to the sequential order and are numbered using the positive integers 1, 2, 3 etc.

The numeric value range for numeric expressions is the set of all integers (i.e. negative, zero and positive integers are permitted).

5.1.3.3 Object identifier expressions

Some attributes that apply to object descriptions have values that contain references to other object descriptions.

In an object class description or a layout style, the value of such an attribute can be represented by an object identifier expression. This is an expression that, when it is evaluated in the proper context, produces the value of an object identifier.

An object identifier expression consists of a reference to an object selection function and the specification of one or two parameters as required by the object selection function concerned.

Object selection functions are

- CURRENT-OBJECT: a function without a parameter, which produces the identifier of the object description to which the attribute applies whose value is represented by the object identifier expression.
- CURRENT-INSTANCE: a function with two parameters, which produces the identifier of that instance of an object of the object class or object type specified by the first parameter that is current relative to the position in the specific structure of the object specified by the second parameter. (This function is explained further in 5.1.3.5).
- SUPERIOR-OBJECT: a function with a parameter, which produces the identifier of the object description that is immediately superior to the object description identified by the parameter. The parameter consists of an object identifier expression.
- PRECEDING-OBJECT: a function with a parameter, which produces the identifier of the object description that immediately precedes the object description specified by the parameter in the sequential order, (see 3.1.2). The parameter consists of an object identifier expression.

Any object selection function will produce a null object identifier when an attempt is made to reference a non-existent object. (For example, the object superior to the document layout root or the document logical root.)

5.1.3.4 References to binding values

Within string and numeric expressions it is possible to refer to the value of a binding within the attribute “bindings” of either the object description to which the attribute having the expression applies or any other specified object description. Such a reference is represented by the combination of a reference to an object description and a binding name (see 5.3.5.3).

The reference to the object description is represented by an expression which when evaluated produces an object identifier.

Such an expression is called a *binding reference expression*. Its format is the same as that of an object identifier expression, defined in 5.1.3.3, but its semantics are as described below.

A binding reference expression consists of either an object identifier or one of the following four functions which, in this context, are called *binding selection functions*:

- CURRENT-OBJECT;
- CURRENT-INSTANCE;
- SUPERIOR;
- PRECEDING.

The parameters of *binding selection functions* as defined in 5.1.3.3, in particular object identifier expressions used as parameters have the semantics described there and are not in turn interpreted as binding selection function applications.

The semantics of the binding selection functions CURRENT-OBJECT and CURRENT-INSTANCE are identical to those of the corresponding object selection functions defined in 5.1.3.3 and 5.1.3.5.

The semantics of the binding selection functions SUPERIOR and PRECEDING are similar to those of the object selection functions SUPERIOR-OBJECT and PRECEDING-OBJECT defined in 5.1.3.3.

The differences are that, if the result of the function is the identifier of an object description that does not have an attribute “bindings” including a binding with the specified binding name, then the invocation of the function is repeated, using the identifier of the failing object description as the function parameter, until an object description is found that has an attribute “bindings” including a binding with the specified binding name.

In the case that the binding selection function is SUPERIOR, the invocation of the function is repeated on increasingly superior object descriptions, starting from the object description immediately superior to the object description specified by the parameter, until either a binding of the specified binding name is located or the document logical root or document layout root is reached.

In the case that the binding selection function is PRECEDING, the invocation of the function is repeated through the object descriptions in the reverse of the sequential order of the objects in the structure, starting from the object description immediately preceding the object description specified by the parameter, until either a binding of the specified binding name is located or the document logical root or document layout root is reached.

If the specified object identifier or the result of the binding reference expression does not pertain to an object that has a binding with the specified binding name, then the result of the reference to the binding value is either an empty string, the integer zero, or a null object identifier, depending on whether a string, a numeric value or an object identifier expression is expected in the context concerned.

The value of the binding which is referenced can contain an expression, which may itself refer to further expressions.

5.1.3.5 *Current instance function*

The CURRENT-INSTANCE function can be used in object identifier expressions (see 5.1.3.3) and in binding reference expressions (see 5.1.3.4). It has two parameters. The first parameter is either an object class identifier or an object type.

The second parameter consists of a reference to a logical object or layout object and is represented by either an object identifier or an object identifier expression.

The result of the function is the object identifier of that instance of the object class or object type specified by the first parameter, that is current relative to the position in the document corresponding to the logical or layout object specified by the second parameter.

To determine the result of the function, four cases are distinguished:

- a) the first parameter specifies a logical object class or logical object type, and the second parameter refers to an object that is part of the specific logical structure (that is, excluding any logical objects generated as a result of the attribute “logical source”);
- b) the first parameter specifies a layout object class or layout object type, and the second parameter specifies a logical object that either has or has not been generated as a result of the attribute “logical source”;
- c) the first parameter specifies a logical object class or logical object type, and the second parameter specifies a temporary logical object that has been generated as a result of the attribute “logical source”;

- d) the first parameter specifies a logical object class or logical object type, and the second parameter refers to a layout object that is of a class that is referenced by at least one basic layout object without generic content.

In case a), the logical object specified by the second parameter is called the “reference logical object” and

- if the reference logical object belongs to the object class or object type specified by the first parameter, then the result of the function is the identifier of that logical object;
- otherwise, the result of the function is the identifier of the nearest superior of the reference logical object that belongs to the object class or object type specified by the first parameter.

In case b), the “reference layout object” is defined as the first basic layout object in which any content is laid out of the logical object specified by the second parameter and

- if the reference layout object belongs to the object class or object type specified by the first parameter, then the result of the function is the identifier of that layout object;
- otherwise, the result of the function is the identifier of the nearest superior of the reference layout object that belongs to the object class or object type specified by the first parameter.

In case c), the “reference layout object” is defined as the first basic layout object in which any content is laid out of the logical object specified by the second parameter and

- the “reference logical object” is defined as the first basic object of the specific logical structure (that is, excluding any temporary logical objects generated as a result of the attribute “logical source”) of which any content is laid out in a layout object that follows the reference layout object in the sequential order;
- if the reference logical object belongs to the object class or object type specified by the first parameter, then the result of the function is the identifier of that logical object;
- otherwise, the result of the function is the identifier of the nearest superior of the reference logical object that belongs to the object class or object type specified by the first parameter.

In case d), the “reference logical object” is defined as the first logical object of which any content is laid out in the layout object referenced by the second parameter and

- if the reference logical object belongs to the object class or object type specified by the first parameter, then the result of the function is the identifier of that logical object;

— otherwise, the result of the function is the identifier of the nearest superior of the reference logical object that belongs to the object class or object type specified by the first parameter.

Any current instance function will produce a null identifier when an attempt is made to reference a non-existent object. [For example, in case a) if neither the reference logical object nor any of its superiors are of the object class or object type specified by the first parameter.]

5.2 Attribute specification format

The attribute definitions in this section are structured as follows:

Constituents:

States the types of constituents for which the attribute can be specified. For layout directives, also states the types of logical components to which the attribute can be applied.

Classification:

States whether the attribute is classified as mandatory, non-mandatory or defaultable, and for which types of constituent.

Structure:

States the structuring of the attribute into parameters and sub-parameters, if any. This entry is omitted if not applicable.

Permissible values:

States the permissible values of the attribute. If the attribute is structured into parameters and sub-parameters the permissible values are specified for these.

Representation:

States the representation of the values in the interchange format. Only stated in certain cases, in general this information is defined in ISO 8613-5.

Default values:

States the default values of the attribute [see 5.1.2.4 j)]. If the attribute is structured into parameters and sub-parameters the default values are specified for these. This entry is omitted if not applicable.

Definition:

Textual description of the semantics of the attribute.

Exceptions:

States any exceptional cases that do not follow the general rules specified for the attribute definition. The rationale for these exceptional cases is issues such as particular optimisations available with restricted usage of the attribute and backwards compatibility with previous standards.

5.3 Shared Attributes

The attributes defined in this clause can be specified for more than one type of constituent. Attributes that may be specified only for logical components, or only for layout components, or only for one type of constituent, are described in subsequent clauses.

5.3.1 Identification attributes

These attributes are used to uniquely identify the component to which they apply.

5.3.1.1 Object type

Constituents:

Component descriptions.

Classification:

Mandatory for all object class descriptions;

Defaultable for an object description which refers to an object class description, otherwise mandatory.

Permissible values:

A set of data elements defined for the attribute.

In the case of a layout component description: “document layout root”; “page set”; “composite or basic page”; “frame”; “block”.

In the case of a logical component description: “document logical root”; “composite logical object”; “basic logical object”.

Default value:

Never applies, since value is always determined by one of the steps a) or c) in 5.1.2.4.

Definition:

This attribute specifies the object type. The object type determines the attributes that may be specified for the object description or object class description.

In the case of a layout object description, the attribute specifies whether the object is of object type

- document layout root;
- page set;
- composite or basic page;
- frame;
- block.

A page is a composite page if it has any subordinates (see 5.3.3.2). It is a basic page if it has no subordinates.

In the case of a logical object description, the attribute specifies whether the object is of object type

- document logical root;
- composite logical object;
- basic logical object.

In the case of an object class description, the attribute specifies the object type of the objects in the object class.

5.3.1.2 *Object identifier*

Constituents:

Object descriptions.

Classification:

Mandatory, unless the exceptional case described below applies, in which case the attribute is non-mandatory.

Permissible values:

A sequence of non-negative integers. The values assigned to the first integer are

- 1 if the constituent is a layout object description;
- 3 if the constituent is a logical object description.

Representation:

A character string consisting of decimal numerals and space characters. The decimal numerals are in one to one correspondence with the integers constituting the identifier: a space character is used as a separator between successive numerals.

Definition:

This attribute identifies an object description uniquely within the context of the document.

An object identifier consists of a sequence of integers. Each integer in this sequence corresponds to a hierarchical level of the specific layout structure or specific logical structure and identifies one particular object description representing an object at that level.

The integers in this sequence start with the integer corresponding to the object description of the document layout root or document logical root. This is followed by each of the integers corresponding to the object descriptions on the path through the hierarchical structure from the document layout root or document logical root to the object description.

The first integer in the sequence indicates whether the identifier pertains to a layout object description or a logical object description.

An object identifier consisting of just a single integer identifies the object description of the document layout root or document logical root.

The actual value of each subsequent integer is not significant; however the sequence of integers allocated to each object description shall be chosen so that each object description can be uniquely distinguished from all other object descriptions in the document.

Exceptions:

This attribute is non-mandatory in certain documents. These documents are those which have all of the following characteristics:

- The interchange format class used for document interchange is class B (see ISO 8613-5), consequently the only specific structure present is the specific layout structure.
- The only object types present in the document are document layout root, pages and blocks.
- There is no use of object identifiers in attributes.

In documents adhering to these rules any two consecutive objects of the same object type in the data stream have the same immediate superior. Thus, under these conditions the semantics of the attribute “object identifier” can be transmitted to the recipient implicitly and the attribute need not be explicitly specified.

NOTE This exceptional case is provided for compatibility with CCITT Recommendations.

5.3.1.3 *Object class identifier*

Constituents:

Object class descriptions.

Classification:

Mandatory.

Permissible values:

A sequence of non-negative integers. The values assigned to the first integer are:

- 0 if the constituent is a layout object class description;
- 2 if the constituent is a logical object class description.

Representation:

A character string consisting of decimal numerals and space characters. The decimal numerals are in one to one correspondence with the integers constituting the identifier: a space character is used as a separator between successive numerals.

Definition:

This attribute identifies an object class description uniquely within the context of the document.

An object class identifier consists of a sequence of integers. The first integer in this sequence indicates whether the identifier pertains to a layout object class description or a logical object class description.

An object class identifier consisting of just this first integer identifies an object class description for the document layout root or document logical root.

The allocation of the other integers is not constrained, other than that the identifier of each object class must be unique.

5.3.2 Construction attributes

These attributes specify rules for controlling the generation of object descriptions from object class descriptions and for controlling the generation of content.

5.3.2.1 Generator for subordinates

Constituents:

Composite object class descriptions.

Classification:

Non-mandatory.

In the case of a complete generator set of logical object class descriptions, this attribute is mandatory for composite logical object class descriptions. In the case of a complete generator set of layout object class descriptions, this attribute is mandatory for all composite layout object class descriptions except those for lowest level frames, for which it is non-mandatory.

In the case of a factor set of object class descriptions, this attribute shall not be specified.

Permissible values:

A construction expression (see definition below):

Definition:

This attribute specifies which objects, and which combinations of objects, may be immediately subordinate to an object of the class. In addition, this attribute specifies an ordering among these immediately subordinate objects.

The value of this attribute is an expression which can be evaluated in a number of ways to yield a set of possible values. Each value is a sequence of object class identifiers representing a sequence of object classes.

If the attribute is present in a composite object class description within a complete generator set of object class descriptions, then its set of possible values specify a constraint for all objects of the class, restricting the permissible immediately subordinate objects.

If the attribute is present in a composite object class description within a partial generator set of object class descriptions, then it does not constrain the immediately subordinate objects for objects of the class. However, it can be used as a guide for creating or editing the specific structure.

If the attribute is specified, then the constraint, in the case of a complete generator set of object class descriptions, or the guide, in the case of a partial generator set of object class descriptions, is as follows.

Each member of a set of object descriptions which has a common immediate superior has a value for the attribute "object class". If a sequence is formed consisting of the values of the attribute "object class" for all object descriptions in the set, in the order specified among those object descriptions by the attribute "subordinates" of their common immediate superior, then this sequence must be one of the values that can be generated by the attribute "generator for subordinates" of their common immediate superior.

If the attribute is absent in a composite object class description, then no constraints are specified for the set of immediately subordinate objects of objects of the class.

The value of this attribute consists of a construction expression. A construction expression specifies the identifiers of object class descriptions that can be used to generate immediately subordinate object descriptions of the object description being generated.

A construction expression is either a construction term (see below) or a construction type. A construction type is one of the following:

- a sequence construction, which consists of one or more construction terms, which are to be evaluated in the order specified;
- an aggregate construction, which consists of one or more construction terms, which are to be evaluated in an arbitrary order;
- a choice construction, which consists of one or more construction terms, one of which is to be evaluated.

A construction term is one of the following:

- a required construction factor;
- an optional construction factor;
- a repetitive construction factor;
- an optional repetitive construction factor.

Each construction factor is either an object class identifier or a construction type. In the former case, the value of the construction factor is the object class identifier. In the latter case, the value of the construction factor is derived by evaluation of the construction type. Evaluation of the construction type may produce either an empty sequence or a sequence of one or more object class identifiers.

A required construction factor is to be evaluated once when the containing construction term is evaluated.

An optional construction factor may be evaluated once or may not be evaluated, when the containing construction term is evaluated.

A repetitive construction factor is to be evaluated one or more times in succession when the containing construction term is evaluated.

An optional repetitive construction factor may be evaluated one or more times in succession, or may not be evaluated, when the containing construction term is evaluated.

5.3.2.2 Content generator

Constituents:

Basic object class descriptions, basic logical object descriptions.

Classification:

Non-mandatory.

Permissible values:

A string expression.

Definition:

The value of this attribute is a string expression, which, when evaluated, produces the content associated with the object. String expressions are defined in **5.1.3.1**.

When a content generator is specified for a logical object class, it is interpreted as providing the default value for this attribute of the logical object descriptions for objects of that object class.

For any component description, this attribute is ignored if there is more than one content portion or if the single one specifies the attribute “content information”.

The attribute “content generator” is evaluated during the layout process and specifies a value for the attribute “content information”, the content portion description, if any, is used to specify other content portion attributes.

When a content generator is evaluated, the resulting content is laid out and presented in accordance with the layout directives and presentation attributes applicable.

The content architecture class of the basic component together with the attributes of the content portion, if any, determine how to interpret the string expression. The string expression may represent character content, raster graphics content or geometric graphics content, with type of coding, coding attributes and alternative representation as defined. Any character string literals in a content generator shall pertain to the character set and control functions specified for the particular content architecture class.

5.3.3 Relationship attributes

These attributes specify the relationships between objects, between objects and object classes, between objects and content portions and between objects and presentation styles.

5.3.3.1 Object class

Constituents:

Object descriptions.

Classification:

Non-mandatory.

In the case of a complete generator set of logical object class descriptions this attribute is mandatory for logical object descriptions. In the case of a complete generator set of layout object class descriptions this attribute is mandatory for composite layout object descriptions.

Permissible values:

The identifier of an object class.

Definition:

This attribute is used to establish a relationship between an object description and its object class description.

The value of this attribute is the identifier of the corresponding object class description (see **5.3.1.3**).

5.3.3.2 Subordinates

Constituents:

Composite object descriptions.

Classification:

Mandatory, unless the exceptional case described below applies, in which case the attribute is non-mandatory.

Permissible values:

A sequence of one or more non-negative integers.

Definition:

This attribute identifies the set of objects immediately subordinate to the object for which this attribute is specified.

The value of this attribute is a sequence of one or more integers. Each integer corresponds to an immediately subordinate object description and consists of the last integer in the identifier of that subordinate object description (see **5.3.1.2**). The sequence contains integers corresponding to each immediately subordinate object description and the same integer may not occur more than once in the sequence.

The order of appearance of the integers in the sequence (not the order of their numeric values) defines the sequential order among the immediately subordinate objects.