TECHNICAL
REPORT

ISO/IEC
TR 9573

First edition
1988-12-01

Information processing — SGML support facilities — Techniques for using SGML

Traitement de l'information — Facilités de support pour SGML — Techniques d'utilisation du SGML

Reference number
ISO/IEC/TR 9573 : 1988 (E)

ISO/IEC Copyright Office, P.O. Box 56, 1211 Geneva 20, Switzerland
Printed in Switzerland

Contents

Figures

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) together form a system for worldwide standardization as a whole. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The main task of a technical committee is to prepare International Standards but in exceptional circumstances, the publication of a technical report of one of the following types may be proposed:

- type 1, when the necessary support within the technical committee cannot be obtained for the publication of an International Standard, despite repeated efforts;
- type 2, when the subject is still under technical development requiring wider exposure;
- type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

ISO/IEC/TR 9573, which is a technical report of type 3, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

This page intentionally left blank

Information processing — SGML support facilities — Techniques for using SGML

0 Introduction

ISO 8879, *Information processing — Text and office systems — Standard Generalized Markup Language (SGML)*, states the rules for the description and markup of documents for their publication and interchange. A basic document type, primarily for computer-assisted publishing, is provided as an example in clause E.1 of ISO 8879, but is not explained there. It is explained in this Technical Report to assist comprehension. Also given is a variety of examples on the use of SGML. Thus this Technical Report is complementary to ISO 8879, its principal purpose being to assist in the adoption of the language.

1 Scope and Field of Application

This Technical Report complements ISO 8879 by providing additional turorial information. It is not intended, and should not be regarded, as an extention, modification, or interpretation of ISO 8879. The SGML language contains a number of components, some of which are optional features. The tutorial information covers the main components of the language only; notably a discussion of LINK, CONCUR, and DATATAG is outside the scope of this Technical Report.

The intended audience is mainly document type designers already familiar with the basic concepts of SGML, but requiring more tutorial information on techniques for using SGML for various applications. Subclauses 5.3 and 8.4 are written in the style of a "User Guide" and can be used as a basis for end-user documentation. For an introductory tutorial on SGML the annexes of ISO 8879 can be used.

This Technical Report includes notes on the analysis of a document prior to the writing of a formal document type definition, and a series of examples.

The principal example is for a general document type, formally defined as an example in clause E.1 of ISO 8879. Others of a general nature are for letter and memorandum, spreadsheet, mathematics, and the mixing of text and graphics. Those for language applications include Scandinavian runes, Japanese, a European multilingual document, and mixing text in languages written from left to right and from right to left.

NOTE — Throughout this Technical Report terms like "is keyed in", and "with keyboarding" are used. This does not neccessarily imply that the markup is to be added explicitly by a user; for text entry one would expect structured, context sensitive, editors to be used, or the markup added by application programs, e.g. in the case of interchange of spreadsheets.

2 References

ISO 646, *Information processing — ISO 7-bit coded character set for information interchange.*

ISO 2022, *Information processing — ISO 7-bit and 8-bit coded character sets — Code extension techniques.*

ISO 4873, *Information processing — ISO 8-bit code for information interchange — Structure and rules for implementation.*

ISO 6937, *Information processing — Coded character sets for text communication.*

ISO 8632-2, *Information processing systems — Computer graphics — Metafile for the storage and transfer of picture description information — Part 2: Character encoding.*

ISO 8632-4, *Information processing systems — Computer graphics — Metafile for the storage and transfer of picture description information — Part 4: Clear text encoding.*

ISO 8859-6, *Information processing — 8-bit single-byte coded graphic character sets — Part 6: Latin/Arabic alphabet.*

ISO 8879, *Information processing — Text and office systems — Standard Generalized Markup Language (SGML).*

ISO 9069, *Information processing — SGML support facilities — SGML Document Interchange Format (SDIF).*[1]

ISO 9070, *Information processing — SGML support facilities — Registration procedures for public text owner identifiers.*[1]

JIS X 0201 (formerly C 6220), *Code for Information Interchange.*

JIS X 0208 (formerly C 6226), *Code of the Japanese Graphic Character Set for Information Interchange.*

3 Definitions

The definitions in ISO 8879 also apply to this Technical Report.

4 Document Analysis for the Creation of a Document Type Definition

4.1 General notes

The analysis of the different classes of documents in an organization that is going to apply SGML is of critical importance for the success of the application. The task of creating a document type definition (DTD) to reflect the structure of an existing class of documents (reports for example) with a view to applying SGML just to continue the production of paper copies poses very few problems. More difficult, but very important, is to analyse the future need of identifying components of documents with a view to processing the documents in a different way, such as creating a database, searchable with standard DBMS software, or multimedia publishing (for example microfiche and CD ROM), or producing secondary publications (for example of titles and abstracts only).

[1] At present at the stage of draft.

4.1.1 Steps of document analysis

The initial step of the analysis is to identify the different Document Type Definitions that are required. One area that requires careful consideration is whether a particular class of documents really has unique elements and structure (and thus requiring a separate DTD), or whether it differs from another class only in the presentation — layout — style. Consider, for example, one document class where the front matter consists of

— one page with title, author's name, and a reference number,

— one page with an abstract,

and another class where part of the first page consists of

— title, author's name, and a reference number,

— an abstract,

followed, on the same page, by the body of the document. In most cases, in spite of the apparent differences, these two classes of documents are really only one document type with two styles of presentation.

The first step, performed for each type of document, is to identify the elements and any attributes required for each element. Techniques that can be applied in this area are discussed in more detail below for some typical parts of documents. In this analysis it is highly desirable to identify the elements that are common to more than one document type, such as paragraphs, footnotes, lists, and headings, in order to make a consistent set of document type definitions, which facilitates the keyboarding and makes it possible to merge parts of documents of different types more easily. Within any organization it is very likely that the body elements of many types of documents are identical, and differences only occur in the front and back matter. Thus the body elements of a letter class may well be the same as those of a report, whereas the "front matter" of a letter would be addressee, sender, date, and reference number, and that of a report, title, author, abstract, and table of contents.

The second step is to identify the hierarchical structure of the elements, starting from the complete document, then dividing it into major elements that in their turn contain other elements, and so on down to the character (text) level.

The third step is to evaluate the use of short references.

Once these three steps have been completed, the formal document type definition can be created with relative ease.

4.1.2 Design issues

In the design of the elements and attributes for a document type, the possibility of text that is automatically generated by the application software/text formatter should be considered. In this way a number of tasks will be facilitated when a document is revised or when parts of several documents are merged. Common cases are listed below:

a) Numbering of elements such as headings, figures, tables, and footnotes. Rather than tagging

```
<head>2.1 Widget Factory
 ...
 ...
described in 2.1 ...
```

it is preferable to design the document type definition in such a way as to tag

```
<head id=widget>Widget Factory
 ...
 ...
described in <hdref refid=widget> ...
```

and let the application software (or text formatter) number the headings and resolve the cross-references. It should be noted that the HDREF tag does not specify the form of the reference and it could have generated "Widget Factory on page 22", or "section 2.1 on page 22" instead. Thus even the form of cross-references is, just as the presentation style, independent of the SGML source document.

b) Creation of the table of contents, and lists of figures and tables. It is preferable that the application software (or text formatter) accumulates the entries and that only the position where the information should be placed in the document is indicated by a tag such as <toc>, <figlist>, and <tablist>. In most document types the position of these elements in the document is always fixed and these tags can be omitted using tag minimization.

c) Creation of an index. It is preferable to be able to insert tags such as

```
<il>words to be put in index</il>
```

in the text and have (correct) page numbers added automatically and the entries collected and sorted by the text formatter and printed in a place indicated by an <index> tag, rather than having to create manually the sorted index entries with appropriate page numbers that have to be updated for any revision of the text.

There are also a number of issues that are technically less important, but that can be very important for the ease of use of an SGML application. These issues include the following:

a) Keystroke minimization in generic identifiers. If the tags are keyed by the author, it is preferable that the generic identifiers for those tags that are not very commonly used are longer and easier to remember rather than being short and perhaps more cryptic. For example <isbn> is preferable to <bn>, even if it costs two more keystrokes. For commonly used tags, for example for paragraphs, short tags are preferable. If the tags are keyed by copy-typists, the keystroke minimization savings may be more important than the generic identifiers being easy to remember. However, a list of generic identifiers and attributes could be placed next to the keyboard for easy reference. If the tags are generated by function keys or similar, the advantages of minimizing keystrokes are only small savings on overheads for storage and transmission. In this Technical Report, generic identifiers are of the form used in clause E.1 of ISO 8879.

b) Tag minimization. With parsers supporting the OMITTAG feature, many tags can be omitted. Thus instead of keyboarding

```
<h1><h1t>Widgets<h1t>
<p>Widgets are ...</p>
</h1>
```

the DTD would be constructed to permit

```
<h1>Widgets
<p>Widgets are ...
```

The design of a document type definition has, however, to strike a good balance, in cases where a tag can only be omitted under certain conditions, between the saving in keystrokes and the extra effort required to learn whether a tag can be omitted in a particular circumstance. One example where tag minimization has been used excessively is lists where the document type definition has been designed to make it possible to omit the tag for the first list item.

```
<ul>
First list item.
<li>Second list item.
<li>Last list item.
</ul>
```

NOTE — The <UL> tag identifies an “unordered list” (see 5.3.10), and <LI> identifies each “list item”.

It should be noted, however, that it is never an error to key a tag that could have been omitted.

c) Attribute minimization. With parsers supporting the SHORTTAG feature, the attribute name can be omitted from the specification of an attribute value if the value is limited to those of a name token group. In order to ease the use, however, the designer has to bear this in mind in order to make the values meaningful. Thus the definition

```
<!--       ELEMENTS    NAME     VALUE      DEFAULT            -->
<!ATTLIST  tabref      page     (page|no)  no
                       refid    IDREF      #REQUIRED           >
```

with the keying

```
<tabref refid=x page>  (or <tabref refid=x page=page>)
```

is preferable to

```
<!--       ELEMENTS    NAME     VALUE      DEFAULT            -->
<!ATTLIST  tabref      page     (yes|no)   no
                       refid    IDREF      #REQUIRED           >
```

with the keying

```
<tabref refid=x yes>  (or <tabref refid=x page=yes>)
```

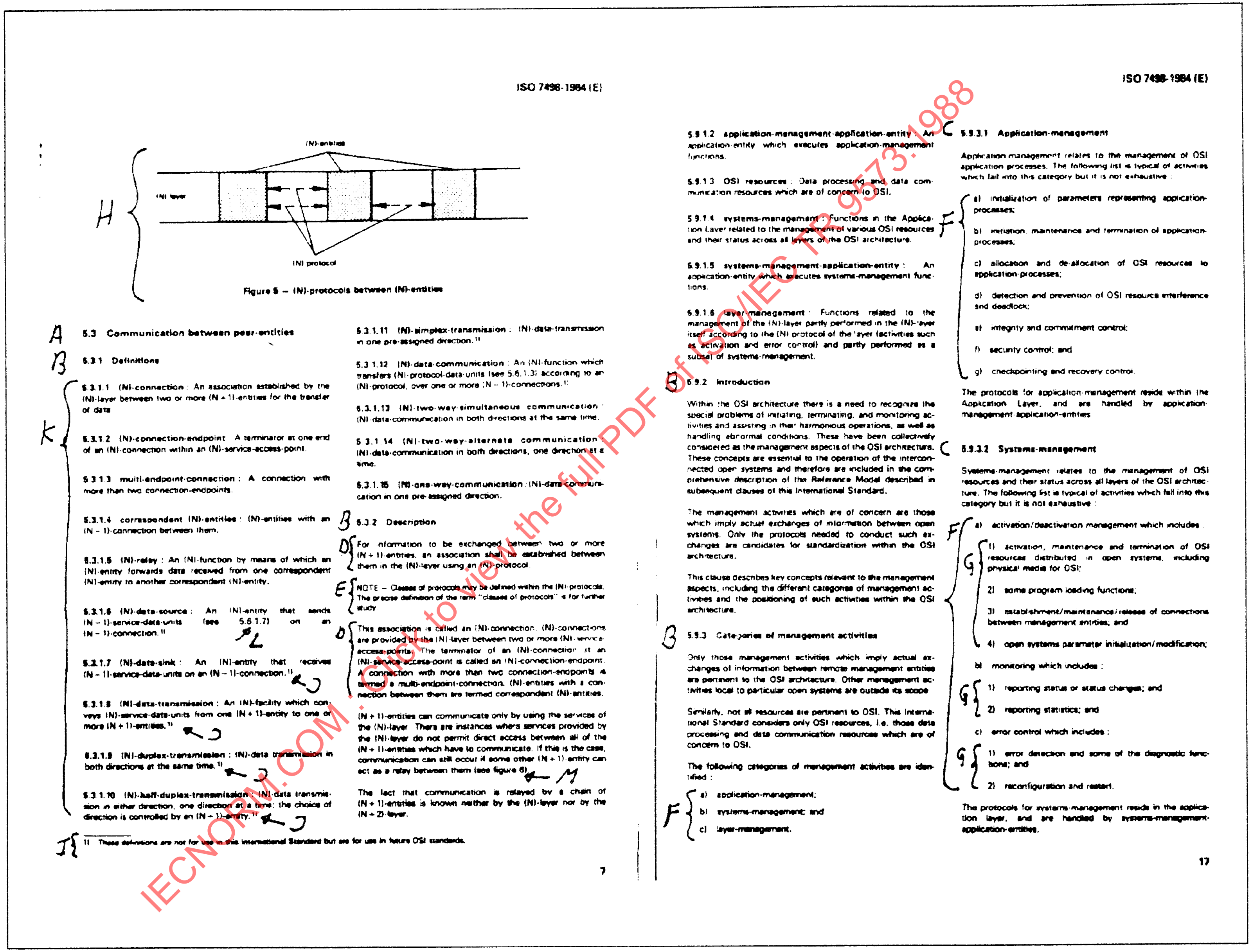
d) Short references. With parsers supporting the SHORTREF feature, some elements may be tagged automatically during the parsing. For example, paired “"” characters could be used as a convenient way of entering <q> and </q> tags. A further discussion of this feature, applied to quotes, is given in clause 14.

Whilst SGML is designed to permit the widest possible variety of applications, it is also clear that if all applications use different tags for common elements, text interchange will not be facilitated by the use of SGML. It is thus hoped that designers of document type definitions will study those publicly available that are of good quality and use the same definitions for elements that are common to both applications.

4.2 Example of the analysis of body text

Consider the pages of body text shown in figure 1. The following elements can readily be identified:

— A, B, and C: headed sections. Call these elements H2, H3, and H4 respectively. An H1 element is also present in other parts of the document.
— D: paragraph. Call this element P.
— E: note. Call this element NOTE.
— F and G: numbered (i.e. ordered), sometimes nested, lists. Call this type of list element OL.
— H: figure. Call this element FIG.
— I: footnote. Call this element FN.
— J: reference to footnote. Call this element FNREF.
— K: list of terms and their definitions, a glossary in fact. Call this element GL.
— L: reference to heading number. Call this element HDREF.
— M: reference to figure. Call this element FIGREF.



A
B
K
L
J
J
J
J
I
B
D
E
D
M
C
F
B
B
F
C
F
G
G
G
H
(N)-entities
(N) layer
(N) protocol

ISO 7498-1984 (E)

Figure 5 – (N)-protocols between (N)-entities

5.3 Communication between peer-entities

5.3.1 Definitions

5.3.1.1 (N)-connection : An association established by the (N)-layer between two or more (N + 1)-entities for the transfer of data

5.3.1.2 (N)-connection-endpoint A terminator at one end of an (N)-connection within an (N)-service-access-point.

5.3.1.3 multi-endpoint-connection : A connection with more than two connection-endpoints.

5.3.1.4 correspondent (N)-entities : (N)-entities with an (N – 1)-connection between them.

5.3.1.5 (N)-relay : An (N)-function by means of which an (N)-entity forwards data received from one correspondent (N)-entity to another correspondent (N)-entity.

5.3.1.6 (N)-data-source : An (N)-entity that sends (N – 1)-service-data-units (see 5.6.1.7) on an (N – 1)-connection.[1]

5.3.1.7 (N)-data-sink : An (N)-entity that receives (N – 1)-service-data-units on an (N – 1)-connection.[1]

5.3.1.8 (N)-data-transmission : An (N)-facility which conveys (N)-service-data-units from one (N + 1)-entity to one or more (N + 1)-entities.[1]

5.3.1.9 (N)-duplex-transmission : (N)-data transmission in both directions at the same time.[1]

5.3.1.10 (N)-half-duplex-transmission : (N)-data transmission in either direction, one direction at a time; the choice of direction is controlled by an (N + 1)-entity.[1]

5.3.1.11 (N)-simplex-transmission : (N)-data-transmission in one pre-assigned direction.[1]

5.3.1.12 (N)-data-communication : An (N)-function which transfers (N)-protocol-data-units (see 5.6.1.3) according to an (N)-protocol, over one or more (N – 1)-connections.[1]

5.3.1.13 (N)-two-way-simultaneous communication : (N)-data-communication in both directions at the same time.

5.3.1.14 (N)-two-way-alternate communication : (N)-data-communication in both directions, one direction at a time.

5.3.1.15 (N)-one-way-communication : (N)-data-communication in one pre-assigned direction.

5.3.2 Description

For information to be exchanged between two or more (N + 1)-entities, an association shall be established between them in the (N)-layer using an (N)-protocol.

NOTE – Classes of protocols may be defined within the (N)-protocols. The precise definition of the term "classes of protocols" is for further study.

This association is called an (N)-connection. (N)-connections are provided by the (N)-layer between two or more (N)-service-access-points. The terminator of an (N)-connection at an (N)-service-access-point is called an (N)-connection-endpoint. A connection with more than two connection-endpoints is termed a multi-endpoint-connection. (N)-entities with a connection between them are termed correspondent (N)-entities.

(N + 1)-entities can communicate only by using the services of the (N)-layer. There are instances where services provided by the (N)-layer do not permit direct access between all of the (N + 1)-entities which have to communicate. If this is the case, communication can still occur if some other (N + 1)-entity can act as a relay between them (see figure 6).

The fact that communication is relayed by a chain of (N + 1)-entities is known neither by the (N)-layer nor by the (N + 2)-layer.

[1] These definitions are not for use in this International Standard but are for use in future OSI standards.

7

ISO 7498-1984 (E)

5.9.1.2 application-management-application-entity : An application-entity which executes application-management functions.

5.9.1.3 OSI resources : Data processing and data communication resources which are of concern to OSI.

5.9.1.4 systems-management : Functions in the Application Layer related to the management of various OSI resources and their status across all layers of the OSI architecture.

5.9.1.5 systems-management-application-entity : An application-entity which executes systems-management functions.

5.9.1.6 layer-management : Functions related to the management of the (N)-layer partly performed in the (N)-layer itself according to the (N) protocol of the layer (activities such as activation and error control) and partly performed as a subset of systems-management.

5.9.2 Introduction

Within the OSI architecture there is a need to recognize the special problems of initiating, terminating, and monitoring activities and assisting in their harmonious operations, as well as handling abnormal conditions. These have been collectively considered as the management aspects of the OSI architecture. These concepts are essential to the operation of the interconnected open systems and therefore are included in the comprehensive description of the Reference Model described in subsequent clauses of this International Standard.

The management activities which are of concern are those which imply actual exchanges of information between open systems. Only the protocols needed to conduct such exchanges are candidates for standardization within the OSI architecture.

This clause describes key concepts relevant to the management aspects, including the different categories of management activities and the positioning of such activities within the OSI architecture.

5.9.3 Categories of management activities

Only those management activities which imply actual exchanges of information between remote management entities are pertinent to the OSI architecture. Other management activities local to particular open systems are outside its scope.

Similarly, not all resources are pertinent to OSI. This International Standard considers only OSI resources, i.e. those data processing and data communication resources which are of concern to OSI.

The following categories of management activities are identified :

a) application-management;

b) systems-management; and

c) layer-management.

5.9.3.1 Application-management

Application management relates to the management of OSI application processes. The following list is typical of activities which fall into this category but it is not exhaustive :

a) initialization of parameters representing application-processes;

b) initiation, maintenance and termination of application-processes;

c) allocation and de-allocation of OSI resources to application-processes;

d) detection and prevention of OSI resource interference and deadlock;

e) integrity and commitment control;

f) security control; and

g) checkpointing and recovery control.

The protocols for application-management reside within the Application Layer, and are handled by application-management-application-entities.

5.9.3.2 Systems-management

Systems-management relates to the management of OSI resources and their status across all layers of the OSI architecture. The following list is typical of activities which fall into this category but it is not exhaustive :

a) activation/deactivation management which includes :

1) activation, maintenance and termination of OSI resources distributed in open systems, including physical media for OSI;

2) some program loading functions;

3) establishment/maintenance/release of connections between management entities; and

4) open systems parameter initialization/modification;

b) monitoring which includes :

1) reporting status or status changes; and

2) reporting statistics; and

c) error control which includes :

1) error detection and some of the diagnostic functions; and

2) reconfiguration and restart.

The protocols for systems-management reside in the application layer, and are handled by systems-management-application-entities.

17

Figure 1 — Sample pages from the body of an International Standard

In order to establish links between the elements that are references and the elements that are referred to, an ID attribute is needed for the heading, figure, and footnote elements, and a REFID attribute is required for the references to heading, figure, and footnote elements.

4.2.1 Headed sections

The headed section elements all have the same structure: a heading title followed by the content of that section. The content of the heading titles is just characters.

NOTE — The hierarchical structure of document elements used in this example means that the content of a headed section lasts until the next headed section of the same level or higher level, rather than ending at the end of the heading title.

Example of keying:

```
<h2>Communication between peer-entities
<h3>Definitions
```

The content of each level is optional and repeatable paragraphs, and similar elements. The parameter entity %s.zz is used for these (defined below), followed by optional and repeatable headings of a lower level. The element definition is thus

```
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT h1            - O  (h1t,(%s.zz;)*,h2*)  -- clause          -->
<!ELEMENT h2            - O  (h2t,(%s.zz;)*,h3*)  -- subclause       -->
<!ELEMENT h3            - O  (h3t,(%s.zz;)*,h4*)  -- subsubclause    -->
<!ELEMENT h4            - O  (h4t,(%s.zz;)*)   -- subsubsubclause    -->
<!ELEMENT (h1t|h2t|h3t|h4t)
                        O O  (#PCDATA)            -- heading title -->
```

NOTES

1 The markup minimization for the elements in this example is discussed in 4.2.10.

2 In many applications there would be no need to differentiate between heading titles of different levels and thus "th" could be used instead of "h1t", "h2t", "h3t", and "h4t".

The content of %s.zz has been identified in this body text as p and note, ol and gl. The following parameter entities can be defined to describe this:

```
<!ENTITY  %  ps.ul.d   "ol"                  -- unit-item lists   -->
<!ENTITY  %  ps.list   "%ps.ul.d;|gl"               -- all lists   -->
<!ENTITY  %  ps.zz     "(%ps.list;)"   -- para/sect subelements   -->
<!ENTITY  %  s.p.d     "p|note"             -- simple paragraphs   -->
<!ENTITY  %  s.zz      "(%s.p.d;)|(%ps.zz;)" -- sect subelements -->
```

NOTE — The multiple levels of parameter entities shown here may at first appear too complex for this simple example of body text. It is shown below, however, that it is very convenient to structure the definition with the use of parameter entities. The naming convention used is explained in 5.5.

4.2.2 Paragraph and note elements

Example of keying of paragraph and note elements:

```
<p>For information to be exchanged between two or more ...
using an (N)-protocol.
<note>Classes of protocols may be defined within the (N)-protocols.
<p>This association ...
```

Paragraphs and notes are very similar in structure and can be defined as

```
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT (%s.p.d;)     O O  %m.p;           -- paragraphs and notes -->
```

The content of these two elements has been identified to be characters, ol, hdref, figref, and fnref. Thus the following can be defined:

```
<!ENTITY  %  p.rf.ph    "hdref|figref"  -- header and figure refs -->
<!ENTITY  %  p.rf.d     "fnref"             -- footnote reference -->
<!ENTITY  %  p.zz.ph    "(%p.rf.ph;)|(%p.rf.d)" -- all references -->
<!ENTITY  %  m.p        "(#PCDATA|(%p.zz.ph;)|(%ps.zz;))*"          >
```

4.2.3 List elements

Two types of list have been identified: an ordered list whose items are ordered and will be numbered in some kind of sequence, and a glossary list whose items comprise a term followed by a definition of that term.

Two approaches to the design of list constructs are described here. In the first approach, each type of list is defined as a different element. In the second, only one (or a very few) forms of list elements are defined and the different kinds of lists are distinguished by attributes. The merits of the two approaches are discussed at the end of this subclause.

4.2.3.1 Separately identified elements

Call the two types of lists: ol and gl. The ol (ordered list) requires an identification of the start and end of the list, an identification of each element in the list (call these li — list item), and a mechanism for nesting the lists. Each list item can contain more than one paragraph. The elements can be defined as

```
<!--      ELEMENTS      MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT (%ps.ul.d;) - -  (li*)            -- unit item lists -->
<!ELEMENT   li          - 0  %m.pseq;                -- list item -->
```

with %m.pseq defined as

```
<!ENTITY  %  m.pseq     "(p,((%s.p.d;)|(%ps.zz;))*)"  -- para seq -->
```

and keyed

```
<ol>
<li>activation/deactivation management which includes:
<ol>
<li>activation, maintenance and termination of OSI resources
distributed in open systems, including physical media for OSI;
...
<li>open systems parameter initialization/modification;
</ol>
<li>monitoring which includes:
...
</ol>
```

NOTE — Another approach to the nesting of lists would be to define different generic identifiers for each level. In the keying of the tags it may thus be clearer which is the current level. With the nesting of different types of lists, however, this advantage is outweighed by the complexity caused by the multitude of generic identifiers required.

The gl (glossary list) is similar to the ordered list, except that each list item is composed of a term followed by its definition. Thus the elements can be defined:

```
<!--      ELEMENTS      MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT gl            - -  ((gt,gd)*)         -- glossary list -->
<!ELEMENT   gt          - 0  (#PCDATA)          -- glossary term -->
<!ELEMENT   gd          - 0  %m.pseq;      -- glossary definition -->
```

NOTE — The parameter entity %m.pseq is re-used, since the content of both the li element and the gd element is the same.

and keyed

```
<gl>
<gt>(N)-connection
<gd>An association established by the (N)-layer between two or
more (N+1)-entities for the transfer of data.
<gt>(N)-connection-endpoint
<gd>A terminator at one end of an (N)-connection within an
(N)-service-access-point.
...
</gl>
```

4.2.3.2 Common list element

In this design, a single list element is defined with one attribute that indicates the form of the list and another that describes whether the list forms any kind of sequence:

```
<!--       ELEMENTS MIN CONTENT                        (EXCEPTIONS) -->
<!ELEMENT list     - -   ((lt?, li)+)                  -- lists -->
<!ELEMENT lt       - O   (#PCDATA)                  -- list term -->
<!ELEMENT li       - O   (%m.pseq;)                 -- list item -->
<!--       ELEMENTS NAME      VALUES                   DEFAULT      -->
<!ATTLIST list     form      (display|gloss)           display
                   seq       (ord|unord)               unord        >
```

With this design, the two types of list might be keyed (and here we show an ordered list within a glossary list, for variety):

```
<list gloss>
<lt>colour
<li>The colour of the transparent background.
This is expressed as three percentages, each representing
a primary colour:
<list ord>
<li>Red
<li>Green
<li>Blue
</list>
<lt>transmissibility
<li>The amount of light that will be transmitted through
the background, expressed as a percentage.
</list>
```

4.2.3.3 Discussion of the two approaches to lists

The first design approach has the advantage of fewer keystrokes, and may be appropriate in a very well-defined or bounded application. The second approach is more suited to general, or evolving, applications:

— Treating the attributes of the lists as true SGML attributes gives flexibility. For example, if one were to provide the option of a glossary in which the items were numbered, then the markup:

```
<list gloss ord>
```

follows naturally from that already described. With the first approach one would need to define a new element, or add an "ordered" attribute to the *gl* element (which would be inconsistent with the use of "ordered" as part of the GI of the *ol* element).

Similarly, a "flowed" list (in which the list is flowed as part of a sentence or paragraph) can be supported by defining *flow* as an alternative to *display* for the *form=* attribute.

— If a document is being edited without the aid of an SGML syntax-directed editing system it is easier to change a list from one type to another if the end-tag does not need alteration.

On the other hand this approach implies that in implementing an SGML syntax-directed editing system it is not sufficient to use the DTD to "prompt" and verify valid sub-elements, since certain combinations of attributes imply that only a subset of the sub-elements permitted by the DTD are actually valid. In a <list unord> element only the <li> sub-elements are valid whereas the DTD also permits <lt> sub-elements to occur.

4.2.4 Footnote element

Footnotes require identification of the start and end of the element, and an optional id attribute for referencing. The element can be defined:

```
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT fn            - -  %m.pseq; -(%i.float;)      -- footnote -->
<!--       ELEMENTS     NAME     VALUE     DEFAULT              -->
<!ATTLIST fn            id       ID        #IMPLIED              >
```

and keyed

```
  ...(N-1)-connection.<fn id=futosi>These definitions are not for use
 in this International Standard but are for use in future OSI standards.
 </fn>
```

NOTE — The -(%i.float;) is explained below.

4.2.5 Figure element

The figure element in this example is composed of a figure body and a figure caption. The figure body is text and/or a drawing. The definition of the element can be

```
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT fig           - -  (figbody,figcap)  -(%i.float;)        >
<!ELEMENT   figbody     0 0  ((artwork|%m.pseq;)*)                 >
<!ELEMENT   figcap      - 0  (#PCDATA)                             >
<!ELEMENT artwork       - 0  EMPTY                                 >
<!--       ELEMENTS     NAME     VALUE     DEFAULT                -->
<!ATTLIST fig           id       ID        #IMPLIED                >
<!ATTLIST artwork       name     ENTITY    #REQUIRED               >
```

and keyed

```
<fig>
<artwork name=nprotcl>
<figcap>(N)-protocol between (N)-entities
</fig>
```

4.2.6 Floating elements

The placement of figures and footnotes is, in general, not fixed; they are floating elements. They are best defined as includable in any place, except inside a figure or footnote. They can thus be defined as

```
<!ENTITY  %  doctype   "standard"                                   >
<!ENTITY  %  i.float   "fig|fn"              -- floating elements -->
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT %doctype      - -  (frontm,body,annex?) +(%i.float;)      >
```

4.2.7 Reference elements

References to headings, figures, and footnotes all have the same structure. Their content is "empty", i.e. it is generated by the text formatter or the application software. These elements can be defined as

```
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT (%p.rf.ph;|%p.rf.d;)
                        - 0  EMPTY           -- generated references -->
<!--       ELEMENTS     NAME      VALUE      DEFAULT                 -->
<!ATTLIST (%p.rf.ph;|%p.rf.d;)
                        refid     IDREF      #REQUIRED                >
```

and keyed

```
 ... a relay between them <figref refid=figxy>.
```

4.2.8 Summary of the definition of the body element

Based on the two pages of body text shown in figure 1, the document type definition for the body components would be

```
<!--           parameter entities                               -->
<!ENTITY  %  ps.ul.d    "ol"                 -- unit-item lists  -->
<!ENTITY  %  ps.list    "%ps.ul.d;|gl"          -- all lists     -->
<!ENTITY  %  ps.zz      "(%ps.list;)"  -- para/sect subelements  -->
<!ENTITY  %  s.p.d      "p|note"           -- simple paragraphs  -->
<!ENTITY  %  s.zz       "(%s.p.d;)|(%ps.zz;)" -- sect subelements -->
<!ENTITY  %  p.rf.ph    "hdref|figref" -- header and figure refs -->
<!ENTITY  %  p.rf.d     "fnref"          -- footnote reference -->
<!ENTITY  %  p.zz.ph    "(%p.rf.ph;)|(%p.rf.d)" -- all references -->
<!ENTITY  %  m.p        "(#PCDATA|(%p.zz.ph;)|(%ps.zz;))*"         >
<!ENTITY  %  m.pseq     "(p,((%s.p.d;)|(%ps.zz;))*)"  -- para seq -->
<!ENTITY  %  i.float    "fig|fn"           -- floating elements -->
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!--            headings                                         -->
<!ELEMENT h1            - 0  (h1t,(%s.zz;)*,h2*)  -- clause      -->
<!ELEMENT h2            - 0  (h2t,(%s.zz;)*,h3*)  -- subclause   -->
<!ELEMENT h3            - 0  (h3t,(%s.zz;)*,h4*)  -- subsubclause -->
<!ELEMENT h4            - 0  (h4t,(%s.zz;)*)    -- subsubsubclause -->
<!ELEMENT (h1t|h2t|h3t|h4t)
                        0 0  (#PCDATA)          -- heading title -->
<!--            paragraphs and notes                             -->
<!ELEMENT (%s.p.d;)     0 0  %m.p;        -- paragraphs and notes -->
<!ELEMENT fn            - -  %m.pseq; -(%i.float;)     -- footnote -->
<!--            lists                                            -->
<!ELEMENT (%ps.ul.d;) - -  (li*)            -- unit item lists -->
<!ELEMENT   li          - 0  %m.pseq;                -- list item -->
<!ELEMENT gl            - -  ((gt,gd)*)          -- glossary list -->
<!ELEMENT   gt          - 0  (#PCDATA)           -- glossary term -->
<!ELEMENT   gd          - 0  %m.pseq;      -- glossary definition -->
<!--            figure                                           -->
<!ELEMENT fig           - -  (figbody,figcap)  -(%i.float;)        >
<!ELEMENT   figbody     0 0  ((artwork|%m.pseq;)*)                 >
<!ELEMENT   figcap      - 0  (#PCDATA)                             >
<!ELEMENT artwork       - 0  EMPTY                                 >
<!--            references                                       -->
<!ELEMENT (%p.rf.ph;|%p.rf.d;)
                        - 0  EMPTY           -- generated references -->
<!--       ELEMENTS     NAME      VALUE      DEFAULT                 -->
<!ATTLIST fn            id        ID         #IMPLIED                 >
<!ATTLIST fig           id        ID         #IMPLIED                 >
```

```
<!ATTLIST artwork      name     ENTITY     #REQUIRED                  >
<!ATTLIST (%p.rf.ph;|%p.rf.d;)
                       refid    IDREF      #REQUIRED                  >
```

4.2.9 Additional elements

A further analysis of more documents of the same type, however, reveals additional elements that need incorporating in the definition, for example

— List of notes. Call this element NL (note list).
— List in which entries are not identified by alphanumeric characters but which are each preceeded by, for example, a dash. Call this element UL (unordered list).
— Examples. Call this element XMP.
— References to headings, figures, tables, and footnotes in external documents. Call this element EXTREF.

The two new types of lists encountered have the same structure as the ol already included. Owing to the use of parameter entities in the definition, the addition of these two elements can be acomplished by merely changing the definition of %ps.ul.d to

```
<!ENTITY  %  ps.ul.d   "ol|ul|nl"          -- unit-item lists    -->
```

The example element on the other hand requires the addition of a new element:

```
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT xmp           - -  %m.pseq;                -- example -->
```

For the references to the elements of external documents, the extref element can be defined:

```
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT extref        - O  EMPTY    -- reference to external doc. -->
<!--       ELEMENTS     NAME     VALUE       DEFAULT                -->
<!ATTLIST extref        docnum   CDATA       #REQUIRED
                        refid    IDREF       #REQUIRED
                        type     (hd|fig|tab|fn)   hd                  >
```

NOTE — To resolve the reference, the application software/text formatter will need to interrogate a database of external references. Similarly, the application software/text formatter is required to update this database with information on all elements with an ID attribute, and the use of any references to other documents, for each document formatted. This information will, in addition, assist in determining the impact of a revision of a document on other documents.

4.2.10 Markup minimization

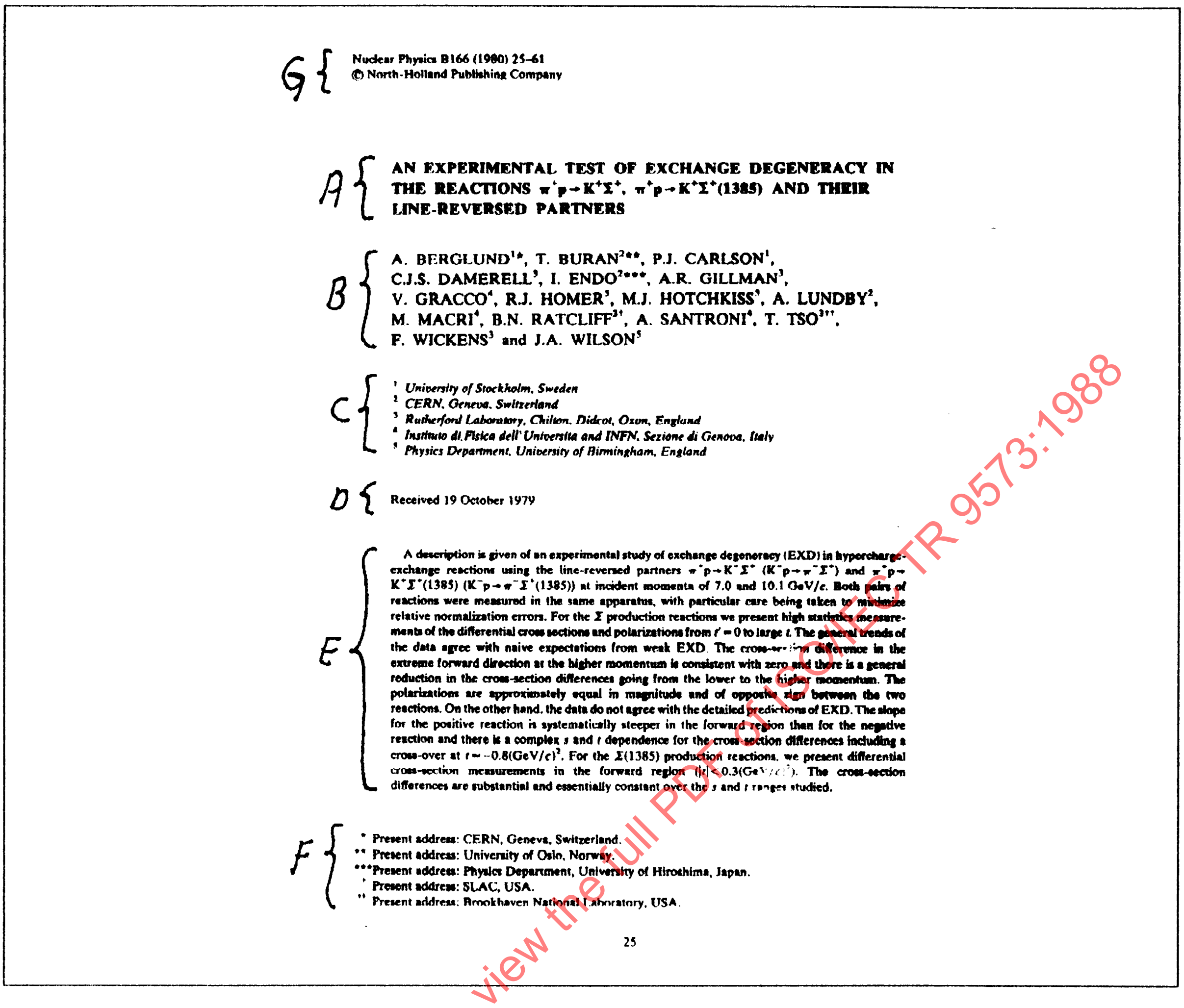
The possibility of markup minimization by the use of OMITTAG is more a question of taste rather than a technical issue. Experience has shown that the definition of tags that can be omitted in this example strikes a good balance between savings in keystrokes, and consistency and ease of learning the omissible tags. It is, however, by no means the only good scheme.

4.3 Example of the analysis of title matter

Consider the title matter in figure 2. Seven main elements can readily be identified: title (A), list of authors (B), list of institutes (C), date (D), abstract (E), notes (F), and publication and copyright information (G). For this kind of document all these elements are required and thus the title matter element can be defined as

```
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT titlem        - -  (title,authlist,instlist,date,abstract) >
```

leaving the notes (F) and publication and copyright information (G) aside for the moment.



Nuclear Physics B166 (1980) 25–61
© North-Holland Publishing Company

AN EXPERIMENTAL TEST OF EXCHANGE DEGENERACY IN THE REACTIONS $\pi^+p\to K^+\Sigma^+$, $\pi^+p\to K^+\Sigma^+(1385)$ AND THEIR LINE-REVERSED PARTNERS

A. BERGLUND[1*], T. BURAN[2**], P.J. CARLSON[1],
C.J.S. DAMERELL[3], I. ENDO[2***], A.R. GILLMAN[3],
V. GRACCO[4], R.J. HOMER[5], M.J. HOTCHKISS[5], A. LUNDBY[2],
M. MACRI[4], B.N. RATCLIFF[3†], A. SANTRONI[4], T. TSO[3††],
F. WICKENS[3] and J.A. WILSON[5]

[1] *University of Stockholm, Sweden*
[2] *CERN, Geneva, Switzerland*
[3] *Rutherford Laboratory, Chilton, Didcot, Oxon, England*
[4] *Instituto di Fisica dell'Universita and INFN, Sezione di Genova, Italy*
[5] *Physics Department, University of Birmingham, England*

Received 19 October 1979

A description is given of an experimental study of exchange degeneracy (EXD) in hypercharge-exchange reactions using the line-reversed partners $\pi^+p\to K^+\Sigma^+$ ($K^-p\to\pi^-\Sigma^+$) and $\pi^+p\to K^+\Sigma^+(1385)$ ($K^-p\to\pi^-\Sigma^+(1385)$) at incident momenta of 7.0 and 10.1 GeV/*c*. Both pairs of reactions were measured in the same apparatus, with particular care being taken to minimize relative normalization errors. For the Σ production reactions we present high statistics measurements of the differential cross sections and polarizations from $t'=0$ to large t. The general trends of the data agree with naive expectations from weak EXD. The cross-section difference in the extreme forward direction at the higher momentum is consistent with zero and there is a general reduction in the cross-section differences going from the lower to the higher momentum. The polarizations are approximately equal in magnitude and of opposite sign between the two reactions. On the other hand, the data do not agree with the detailed predictions of EXD. The slope for the positive reaction is systematically steeper in the forward region than for the negative reaction and there is a complex s and t dependence for the cross-section differences including a cross-over at $t\approx-0.8(\text{GeV}/c)^2$. For the $\Sigma(1385)$ production reactions, we present differential cross-section measurements in the forward region ($|t|<0.3(\text{GeV}/c)^2$). The cross-section differences are substantial and essentially constant over the s and t ranges studied.

[*] Present address: CERN, Geneva, Switzerland.
[**] Present address: University of Oslo, Norway.
[***] Present address: Physics Department, University of Hiroshima, Japan.
[†] Present address: SLAC, USA.
[††] Present address: Brookhaven National Laboratory, USA.

25

Figure 2 — Sample title page

4.3.1 Title element

A

AN EXPERIMENTAL TEST OF EXCHANGE DEGENERACY IN THE REACTIONS $\pi^+p\to K^+\Sigma^+$, $\pi^+p\to K^+\Sigma^+(1385)$ AND THEIR LINE-REVERSED PARTNERS

If the aim is only to produce a printed title page, the definition of the title element could simply be

```
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT title       - -  (#PCDATA)             >
```

and keyed

<title>An Experimental Test of Exchange Degeneracy in the Reactions $\pi^+p\to K^+\Sigma^+$, $\pi^+p\to K^+\Sigma^+(1385)$ and their Line-Reversed Partners
</title>

If the aim is also to make a database of articles or a secondary publication of an index to a series of articles, there may be a need to identify the reactions studied, and the element definitions might be

```
<!--        ELEMENTS    MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT title         - -  ((#PCDATA|reaction)+)  >
<!ELEMENT reaction      - -  (#PCDATA)              >
```

and keyed

<title>An Experimental Test of Exchange Degeneracy in the Reactions
<reaction>$\pi^+ p \rightarrow K^+ \Sigma^+$**</reaction>**, **<reaction>**$\pi^+ p \rightarrow K^+ \Sigma^+ (1385)$**</reaction>**
and their Line-Reversed Partners
</title>

4.3.2 Instlist element

C

[1] *University of Stockholm, Sweden*
[2] *CERN, Geneva, Switzerland*
[3] *Rutherford Laboratory, Chilton, Didcot, Oxon, England*
[4] *Instituto di Fisica dell'Universita and INFN, Sezione di Genova, Italy*
[5] *Physics Department, University of Birmingham, England*

Similarly the list of institutes may require different degrees of structure depending on the application. Two extremes are

```
<!--        ELEMENTS    MIN  CONTENT (EXCEPTIONS)               -->
<!ELEMENT instlist      - -  (inst+) -- list of institutes      -->
<!ELEMENT inst          - O  (#PCDATA) -- institute             -->
<!--        ELEMENTS    NAME     VALUE      DEFAULT             -->
<!ATTLIST inst          id       ID         #IMPLIED             >
```

which is keyed

```
<instlist>
<inst id=usto>University of Stockholm, Sweden.
 ...
<inst id=ubir>Physics Department, University of Birmingham, England.
</instlist>
```

and

```
<!--        ELEMENTS    MIN  CONTENT (EXCEPTIONS)               -->
<!ELEMENT instlist      - -  (inst+) -- list of institutes      -->
<!ELEMENT inst          - O  (dept* , instname* , instqual* ,
                                   city* , county* , country*)   >
<!ELEMENT dept          - O   (#PCDATA) -- department           -->
<!ELEMENT instname      - O   (#PCDATA) -- name of institute    -->
<!ELEMENT instqual      - O   (#PCDATA) -- qualifier            -->
<!ELEMENT city          - O   (#PCDATA) -- city                 -->
<!ELEMENT county        - O   (#PCDATA) -- county               -->
<!ELEMENT country       - O   (#PCDATA) -- country              -->
<!--        ELEMENTS    NAME     VALUE      DEFAULT             -->
<!ATTLIST inst          id       ID         #IMPLIED             >
```

which is keyed

```
<instlist>
<inst id=usto><instname>University of Stockholm<country>Sweden
<inst id=cern><instname>CERN<city>Geneva<country>Switzerland
<inst id=lral><instname>Rutherford Laboratory<city>Chilton
<city>Didcot<county>Oxon<country>England
<inst id=ugen><instname>Instituto di Fisica dell'Universita and INFN
<instqual>Sezione di Genova<country>Italy
<inst id=ubir><dept>Physics Department
<instname>University of Birmingham<country>England
</instlist>
```

NOTE — The data entry for a list of this complexity is clearly well suited for being menu driven, with fields for the various elements of the institute information.

4.3.3 Authlist element

B

A. BERGLUND[1*], T. BURAN[2**], P.J. CARLSON[1],
C.J.S. DAMERELL[3], I. ENDO[2***], A.R. GILLMAN[3],
V. GRACCO[4], R.J. HOMER[5], M.J. HOTCHKISS[3], A. LUNDBY[2],
M. MACRI[4], B.N. RATCLIFF[3†], A. SANTRONI[4], T. TSO[3††],
F. WICKENS[3] and J.A. WILSON[5]

F

[*] Present address: CERN, Geneva, Switzerland.
[**] Present address: University of Oslo, Norway.
[***] Present address: Physics Department, University of Hiroshima, Japan.
[†] Present address: SLAC, USA.
[††] Present address: Brookhaven National Laboratory, USA.

The list of authors contains, in addition to the names, references to institutes and in many cases also to explanatory notes. Since each note is almost always associated with one author, this element is considered part of the authors' list, but placed at the bottom of the page by the text formatter. The authors' names, with a view to database applications, are divided into initials and surname. Thus the definition is

```
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)            -->
<!ELEMENT authlist    - -  ((author|anote)+)                 >
<!ELEMENT author      - O  (initials,surname)                >
<!ELEMENT initials    O O  (#PCDATA)                         >
<!ELEMENT surname     - O  (#PCDATA)                         >
<!ELEMENT anote       - O  (#PCDATA)                         >
<!--      ELEMENTS    NAME      VALUE     DEFAULT           -->
<!ATTLIST author      refid     IDREFS    #REQUIRED          >
<!ATTLIST anote       id        ID        #IMPLIED           >
```

and is keyed

```
<authlist>
<author refid="usto pacern">A<surname>Berglund
<anote id=pacern>Present address: CERN, Geneva, Switzerland.
 ...
<author refid=lral>F<surname>Wickens
<author refid=ubir>JA<surname>Wilson
 ...
</authlist>
```

The explanatory notes of different kind, e.g. "Present address", and "Paid in part by", could be considered as different elements with the definition

```
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)             -->
<!ELEMENT panote      - 0  (#PCDATA)                          >
<!ELEMENT ppnote      - 0  (#PCDATA)                          >
<!--      ELEMENTS        NAME     VALUE     DEFAULT        -->
<!ATTLIST (panote|ppnote) id       ID        #IMPLIED         >
```

and keyed

```
<panote id=pacern>CERN, Geneva, Switzerland
<ppnote id=ppslac>SLAC, USA
```

Alternatively the type could be indicated by an attribute and the element defined

```
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)             -->
<!ELEMENT anote       - 0  (#PCDATA)                          >
<!--      ELEMENTS    NAME     VALUE     DEFAULT            -->
<!ATTLIST anote       id       ID        #IMPLIED
                      type     (pa|pp|none) none              >
```

and keyed

```
<anote id=pacern type=pa>CERN, Geneva, Switzerland
<anote id=ppslac type=pp>SLAC, USA
```

4.3.4 Date element

The date in this journal article can be defined as

```
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)             -->
<!ELEMENT date        - 0  (#PCDATA)                          >
```

and keyed

```
<date>19 October 1979
```

In other cases, different dates of different types, may be necessary, for example

```
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT titlem      - -  (title,authlist,instlist,date+,abstract) >
<!ELEMENT date        - 0  (#PCDATA)                              >
<!--      ELEMENTS    NAME     VALUE                   DEFAULT -->
<!ATTLIST date        type     (receive|publish|revise)   publish   >
```

and keyed

```
<date type=receive>19 October 1979
<date type=publish>20 January 1980
<date type=revise >15 March 1981
<date type=publish>20 May 1981
<date type=revise >30 November 1982
<date type=publish>3 February 1983
```

or separate elements may be defined for the different types of dates

```
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT titlem      - -  (title,authlist,instlist,
                            (recdate|pubdate|revdate)*,abstract)    >
<!ELEMENT recdate     - 0  (#PCDATA)                              >
<!ELEMENT pubdate     - 0  (#PCDATA)                              >
<!ELEMENT revdate     - 0  (#PCDATA)                              >
```

and keyed

```
<recdate>19 October 1979
<pubdate>20 January 1980
<revdate>15 March 1981
<pubdate>20 May 1981
<revdate>30 November 1982
<pubdate>3 February 1983
```

In some cases the date can be defined as a simple character string. In other applications there may be a requirement to identify the day, month, and year as well as the calendar. For example

```
<!--       ELEMENTS    MIN  CONTENT (EXCEPTIONS)             -->
<!ELEMENT  date        - 0  (day,month,year)                  >
<!ELEMENT  day         - 0  (#PCDATA)                         >
<!ELEMENT  month       - 0  (#PCDATA)                         >
<!ELEMENT  year        - 0  (#PCDATA)                         >
<!--       ELEMENTS    NAME     VALUE     DEFAULT            -->
<!ATTLIST  date        calendar (gregor|julian|hegira|mundi)
                                          gregor              >
```

which is keyed

```
<date calendar=gregor><day>4<month>May<year>1987
<date calendar=julian><day>12<month>February<year>1492
<date calendar=mundi><day>23<month>Adar<year>5744
<date calendar=hegira><day>10<month>Ramadan<year>1370
```

In some applications it may be desirable that the application software (or text formatter) generates the current date, rather than the date being always keyboarded.

4.3.5 Abstract element

The abstract contains, in this case, paragraphs of text that consist only of characters. In a more general case paragraphs contain many more subelements that are described in clause 5.

```
<!--       ELEMENTS    MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT  abstract    - -  (p+)                  >
<!ELEMENT  p           - 0  (#PCDATA)             >
```

and keyed

```
<abstract><p>A description is given ... ranges studied.
</abstract>
```

4.3.6 Article element

The title matter under consideration is of a journal article. For the discussion of the publication and copyright information there is a need to look at the element one level higher than the titlem element. This element, article, is composed of publication information, copyright, and a keyword list, followed by titlem and the body of the article. The content of the publication information is automatically generated by the application software (or text formatter) based on information one level higher than the article element. Similarly the copyright notice is also automatically generated. Thus

```
<!--       ELEMENTS    MIN  CONTENT (EXCEPTIONS)          -->
<!ELEMENT  article     - -  (publinfo,copyrigt,keywlist,
                                 titlem,body)              >
<!ELEMENT  publinfo    - 0  EMPTY      -- generated       -->
<!ELEMENT  copyrigt    - 0  EMPTY      -- generated       -->
<!ELEMENT  keywlist    - -  (keyw+)                        >
<!ELEMENT  keyw        - 0  (#PCDATA)                      >
```

4.3.7 Complete definition of the title matter

A full definition of the title matter could thus be

```
<!--       ELEMENTS    MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT article      - -  (publinfo,copyrigt,keywlist,
                                  titlem,body)                    >
<!ELEMENT publinfo     - O  EMPTY      -- generated              -->
<!ELEMENT copyrigt     - O  EMPTY      -- generated              -->
<!ELEMENT keywlist     - -  (keyw+)                               >
<!ELEMENT keyw         - O  (#PCDATA)                             >
<!ELEMENT titlem       - -  (title,authlist,instlist,date,abstract) >
<!ELEMENT title        - -  ((#PCDATA|reaction)+)                 >
<!ELEMENT reaction     - -  (#PCDATA)                             >
<!ELEMENT authlist     - -  ((author|anote)+)                     >
<!ELEMENT author       - O  (initials,surname)                    >
<!ELEMENT initials     O O  (#PCDATA)                             >
<!ELEMENT surname      - O  (#PCDATA)                             >
<!ELEMENT anote        - O  (#PCDATA)                             >
<!ELEMENT instlist     - -  (inst+) -- list of institutes        -->
<!ELEMENT inst         - O  (#PCDATA) -- institute               -->
<!ELEMENT date         - O  (#PCDATA)                             >
<!ELEMENT abstract     - -  (p+)                                  >
<!ELEMENT p            - O  (#PCDATA)  -- paragraph              -->
<!--       ELEMENTS    NAME     VALUE     DEFAULT                -->
<!ATTLIST inst         id       ID        #IMPLIED                >
<!ATTLIST author       refid    IDREFS    #REQUIRED               >
<!ATTLIST anote        id       ID        #IMPLIED                >
```

and keyed

```
<article>
<publinfo>
<copyrigt>
<keywlist>
<keyw>exchange degeneracy
<keyw>line reversal
<keyw>hypercharge exchange
</keywlist>
<titlem>
<title>An Experimental Test of Exchange Degeneracy in the Reactions
<reaction>π⁺p→K⁺Σ⁺</reaction>, <reaction>π⁺p→K⁺Σ⁺(1385)</reaction>
and their Line-reversed Partners
</title>
<authlist>
<author refid="usto pacern">A<surname>Berglund
 ...
<author refid=lral>F<surname>Wickens
<author refid=ubir>JA<surname>Wilson
<anote id=pacern>Present address: CERN, Geneva, Switzerland.
 ...
</authlist>
<instlist>
<inst id=usto>University of Stockholm, Sweden.
 ...
<inst id=lral>Rutherford Laboratory, Chilton, Didcot, Oxon, England.
<inst id=ubir>Physics Department, University of Birmingham, England.
</instlist>
<date>19 October 1979
<abstract><p>A description is given ... ranges studied.
</abstract>
</titlem>
```

```
<body>
 . . .
```

4.4 Example of the analysis of tabular material

Consider the following table:

Grandeur de filetage		Charge d'essai, % Tolérances du filetage		
au-dessus de	jusqu'à	6H	7H	6G
—	M2,5	100	—	95,5
M2,5	M7	100	95,5	97
M7	M16	100	96	97,5
M16	M39	100	98	98,7

One approach would be to consider the table as a "tree structure" being composed of two headed columns. The first column further contains two headed columns, each with four rows, and the second column three headed columns, each with four rows. The element definition would be

```
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS)            -->
<!ELEMENT  table        - -  (col+)                            >
<!ELEMENT  col          - -  (head?,(row+|col+))               >
<!ELEMENT  row          - 0  (#PCDATA)                         >
<!ELEMENT  head         - 0  (#PCDATA,subhead?)                >
<!ELEMENT  subhead      - 0  (#PCDATA)                         >
```

and the example table keyed

```
<table>
<col><head>Grandeur de filetage
<col><head>au-dessus de <row> - <row> M2,5 <row> M7 <row> M16 </col>
<col><head>jusqu'à <row> M2,5 <row> M7 <row> M16 <row> M39 </col>
</col>
<col><head>Charge d'essai, % <subhead> Tolérances du filetage
<col><head> 6H <row> 100 <row> 100 <row> 100 <row> 100 </col>
<col><head> 7H <row>  - <row> 95,5 <row>  96 <row>  98 </col>
<col><head> 6G <row> 95,5 <row> 97 <row> 97,5 <row> 98,7 </col>
</col>
</table>
```

Another approach would be to consider the "tree structure" as being composed of a row of headings, each of two levels, followed by rows of table entries. The element definition would be

```
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS)            -->
<!ELEMENT  table        - -  (head1+,row+)                     >
<!ELEMENT  head1        - 0  (#PCDATA,subhead?,head2*)         >
<!ELEMENT  head2        - 0  (#PCDATA,subhead?)                >
<!ELEMENT  subhead      - 0  (#PCDATA)                         >
<!ELEMENT  row          - 0  (col+)                            >
<!ELEMENT  col          - 0  (#PCDATA)                         >
```

and the example table keyed

```
<table>
<head1>Grandeur de filetage
<head2>au-dessus de <head2>jusqu'à
<head1>Charge d'essai, % <subhead> Tolérances du filetage
<head2> 6H <head2> 7H <head2> 6G
<row><col>  -   <col> M2,5 <col> 100 <col>  -   <col> 95,5
<row><col> M2,5 <col> M7   <col> 100 <col> 95,5 <col> 97
<row><col> M7   <col> M16  <col> 100 <col> 96   <col> 97,5
<row><col> M16  <col> M39  <col> 100 <col> 98   <col> 98,7
</table>
```

NOTE — An element definition for tables based on this approach is given in clause 9.

Another approach would be to describe it "geometrically" as a series of rows, each containing two or five columns, the widths indicated by attributes. Rules are explicitly indicated by tags and attributes. The first two rows are "headers" of two levels. The element definition would be

```
<!--       ELEMENTS      MIN  CONTENT (EXCEPTIONS)               -->
<!ELEMENT  table         - -  (head1,head2?,row+)                 >
<!ELEMENT  (head1|head2) - O  (colh+,hbar?)                       >
<!ELEMENT  colh          - O  (#PCDATA,subhead?)                  >
<!ELEMENT  subhead       - O  (#PCDATA)                           >
<!ELEMENT  hbar          - O  EMPTY                               >
<!ELEMENT  row           - O  (col+,hbar)                         >
<!ELEMENT  col           - O  (#PCDATA)                           >
<!--       ELEMENTS      NAME     VALUE       DEFAULT            -->
<!ATTLIST  table         id       ID          #IMPLIED
                         width    CDATA       #REQUIRED
                         frame    (framed|none) none              >
<!ATTLIST  (head1|head2|row) widths  CDATA       #CURRENT
                         vbars    (on|off)    off                 >
```

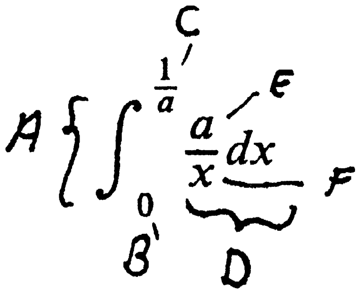
and the example table keyed

```
<table width="87mm" framed>
<head1 widths="36mm 51mm" vbars=on>
<colh>Grandeur de filetage
<colh>Charge d'essai, %<subhead>Tolérances du filetage
<head2 widths="18mm 18mm 17mm 17mm 17mm" vbars=on>
<colh>au-dessus de <colh>jusqu'à<colh>6H<colh>7H<colh>6G
<hbar>
<row widths="18mm 18mm 17mm 17mm 17mm" vbars=on>
<col>  -   <col> M2,5 <col> 100 <col>  -   <col> 95,5
<hbar>
<row vbars=on>
<col> M2,5 <col> M7   <col> 100 <col> 95,5 <col> 97
<hbar>
<row vbars=on>
<col> M7   <col> M16  <col> 100 <col> 96   <col> 97,5
<hbar>
<row vbars=on>
<col> M16  <col> M39  <col> 100 <col> 98   <col> 98,7
</table>
```

NOTE — Layout information would normally rarely be indicated in the tags. Tables, however, is one example where layout parameters cannot always be determined by the application. Using parsers not supporting the LINK feature these have to be specified using attributes, whereas LINK would normally be used if a parser supporting LINK is available.

4.5 Example of the analysis of a simple mathematical formula

For the description of mathematical formulæ a number of “languages” have been proposed and implemented. The basic information that any such description needs to convey is the identification of the components of the formula and their relationships (that is structure). Mathematics is thus well suited to be described by an SGML application. Consider, for example



The components that need identification are

— A: the operator. Call this element INTEGRAL.
— B and C: the lower and upper limit. Call these elements FROM and TO.
— D: the integrand. Call this element OF.

The element definition of the integral can thus be

```
<!--       ELEMENTS      MIN  CONTENT (EXCEPTIONS)                  -->
<!ELEMENT  integral      - -  (from,to,of)                           >
```

The integrand is, in this example, a fraction composed of a numerator and a denominator. The fraction elements can be defined as

```
<!ENTITY % f_oper "integral|frac"  -- partial list of operations -->
<!--       ELEMENTS      MIN  CONTENT (EXCEPTIONS)                  -->
<!ELEMENT  frac          - -  (numer,over)                           >
<!ELEMENT    numer       0 0  (#PCDATA|%f_oper)+                     >
<!ELEMENT    over        - 0  (#PCDATA|%f_oper)+                     >
```

and keyed

```
<frac> a <over> x </frac>
```

In the example, the numerator and denominator are both just #PCDATA, but in a more general case they can be #PCDATA and/or any other operation, such as another fraction. Thus the content has been defined as (#PCDATA|%f_oper;)+. Similarly, the upper and lower limits can take either #PCDATA and/or other operations as content.

The complete definition of the elements for the integral and the fraction is thus

```
<!ENTITY % f_oper "integral|frac"  -- partial list of operations -->
<!--       ELEMENTS      MIN  CONTENT (EXCEPTIONS)                  -->
<!ELEMENT  integral      - -  (from,to,of)                           >
<!ELEMENT  (from|to|of)  - 0  (#PCDATA|%f_oper)+                     >
<!ELEMENT  frac          - -  (numer,over)                           >
<!ELEMENT    numer       0 0  (#PCDATA|%f_oper)+                     >
<!ELEMENT    over        - 0  (#PCDATA|%f_oper)+                     >
```

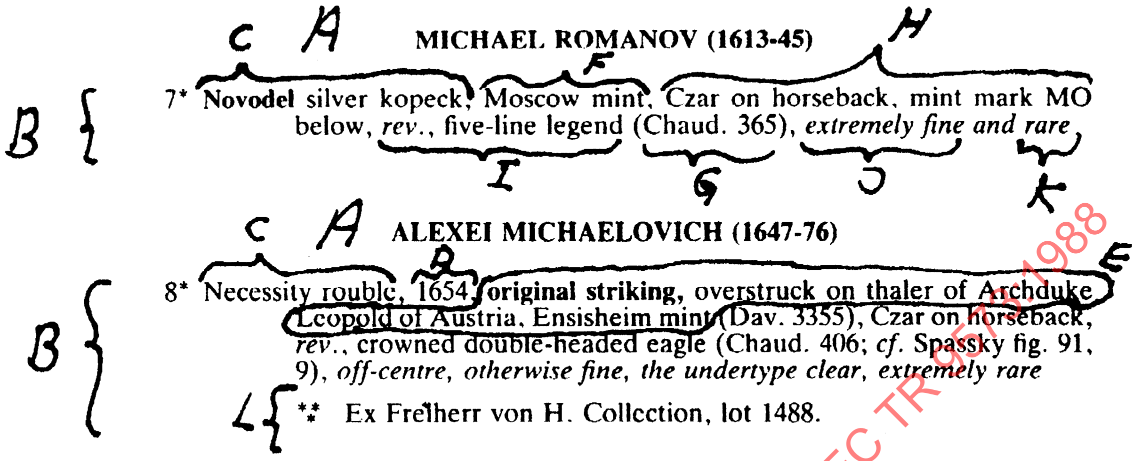
and this example keyed

```
<integral><from> 0 <to> <frac> 1 <over> a </frac> <of>
<frac> a <over> x </frac> dx </integral>
```

NOTE — A complete element definition for mathematics is given in clause 8.

4.6 Example of the analysis of an auction catalogue

Consider the following entries from an auction catalogue:


C
A
H
F
B
I
G
J
K
C
A
D
E
B
L

MICHAEL ROMANOV (1613-45)

7* **Novodel** silver kopeck, Moscow mint, Czar on horseback, mint mark MO below, *rev.*, five-line legend (Chaud. 365), *extremely fine and rare*

ALEXEI MICHAELOVICH (1647-76)

8* Necessity rouble, 1654, **original striking,** overstruck on thaler of Archduke Leopold of Austria, Ensisheim mint (Dav. 3355), Czar on horseback, *rev.*, crowned double-headed eagle (Chaud. 406; *cf.* Spassky fig. 91, 9), *off-centre, otherwise fine, the undertype clear, extremely rare*

** Ex Freiherr von H. Collection, lot 1488.

The following main elements can be identified:

— A: header (in this case the name of a ruler), that can be followed by explanatory text. Call these elements H2 and P.

NOTE — H1 is used to separate the catalogue into countries.

— B: lot description. Call this element LOT. Lots that are illustrated are shown with an asterisk after the number, so an attribute to specify this is needed. Call this attribute ILL.

Each LOT element is divided into several components:

— C: denomination. Call this element DENOM.
— D: date. Call this element DATE.
— E: general comment. Call this element GENCOM.
— F: mint. Call this element MINT.
— G: reference. Call this element REF.
— H: description of obverse. Call this element OBV.
— I: description of reverse. Call this element REV.
— J: condition. Call this element GRADE.
— K: rarity. Call this element RARITY.
— L: provenance. Call this element PROVEN.

For some entries, of less valuable items, the description is shorter. For this reason a number of the elements are optional.

The elements for these catalogue entries can be defined as

```
<!ENTITY  %   p.em.ph    "hp1|hp2|hp3"    -- emphasized phrases    -->
<!--          ELEMENTS    MIN  CONTENT (EXCEPTIONS)                -->
<!ELEMENT h2              - 0  (h2t,p*,lot+)                        >
<!ELEMENT   h2t           0 0  (#PCDATA)                            >
<!ELEMENT p               0 0  ((#PCDATA|%p.em.ph)+)                >
<!ELEMENT lot             - 0  (denom,date?,gencom?,mint?,(obv,rev)?,
                                ref*,grade,rarity?,proven?)         >
<!ELEMENT denom           - 0  ((#PCDATA|%p.em.ph)*)                >
<!ELEMENT date            - 0  (#PCDATA)                            >
<!ELEMENT gencom          - 0  ((#PCDATA|ref|%p.em.ph)*)            >
<!ELEMENT (obv|rev)       - 0  ((#PCDATA|%p.em.ph)*)                >
<!ELEMENT ref             - 0  ((#PCDATA|%p.em.ph)*)                >
```

```
<!ELEMENT mint         - O  ((#PCDATA|%p.em.ph)*)                   >
<!ELEMENT grade        - O  ((#PCDATA|%p.em.ph)*)                   >
<!ELEMENT rarity       - O  ((#PCDATA|%p.em.ph)*)                   >
<!ELEMENT proven       - O  (p+)                                    >
<!ELEMENT (hp1|hp2|hp3) - O (#PCDATA)                               >
<!--      ELEMENTS      NAME     VALUE    DEFAULT                 -->
<!ATTLIST lot           ill      (yes|no) no                        >
```

and these two entries keyed

```
<h2>Michael Romanov (1613-45)
<lot ill=yes><denom><hp2>Novodel</hp2> silver kopek <mint>Moscow
<obv>Czar on horseback, mint mark MO below<rev>five-line legend
<ref>Chaud. 365 <grade>extremely fine<rarity>rare
<h2>Alexei Michaelovich (1647-76)
<lot ill=yes><denom>Necessity Rouble<date>1654<gencom><hp2>original
striking,</hp2> overstruck on thaler of Archduke Leopold of Austria,
Ensisheim mint<ref>Dav 3355.
<obv>Czar on horseback<rev>crowned double-headed eagle
<ref>Chaud. 406; <hp1>cf</hp1> Spassky fig. 91, 9
<grade>off centre, otherwise fine, the undertype clear
<rarity>extremely rare<proven>Freiherr von H. Collection, lot 1488.
```

4.7 The DTD with relation to the size of a document

In general, the body elements described in clause 5 are common to most document classes within an organization. Some elements may not be frequently used in short documents; figures and tables may occur rarely in letters. From the point of view of consistency, and thus easier learning, there appears to be no compelling reason to exclude such elements from the document type definition of shorter documents. On the other hand in some systems there could be a performance implication as the complexity of the application program or text formatter could be affected. In any event, the layout — presentation — style normally does vary with document length and document type.

The front and back matter elements, on the other hand, show, in general, many differences depending on document type and thus average length.

In some organizations there is a requirement for special elements, for example mathematical formulæ. Such elements, however, are likely to be required in all document types regardless of document length. In other words, the element repertoire required is more likely to be a function of the content of the documents in an organization rather than a function of the average length of the documents of each type.

5 General Document

5.1 General notes

The aim of this clause is to provide an example of a practical document type definition for a "general" document that can be used as a basis for application designers when developing applications. It contains all the elements necessary for many applications, and has been constructed in such a way that other elements can be added if the application so requires.

5.2 Document structure

A general document element **(general)** consists of front matter **(frontm)** (optional), followed by the main body **(body)**, followed by an appendix section **(appendix)** (optional), followed by back matter **(backm)**. Both the start and end-tag of the element are required. Index entries **(ix)** and "floating elements" **(%i.float)** (figures and footnotes) are included elements, and these elements can thus be interspersed anywhere in the document.

```
general---frontm----titlep----title-----tline-----%m.ph  ==>
        |         |         |-docnum----#PCDATA
        |         |         |-date------#PCDATA
        |         |         |-abstract--h1t-------%m.ph  ==>
        |         |         |         |-%s.zz  ==>
        |         |         |         |-h2  ==>
        |         |         |
        |         |         |-author----#PCDATA
        |         |         |-address--aline-----%m.ph  ==>
        |         |         |-%s.zz  ==>
        |         |
        |         |-%fm.d----abstract--h1t-------%m.ph  ==>
        |         |         |         |-%s.zz  ==>
        |         |         |         |-h2  ==>
        |         |         |
        |         |         |-preface--h1t-------%m.ph  ==>
        |         |                   |-%s.zz  ==>
        |         |                   |-h2  ==>
        |         |
        |         |-h1--------h1t-------%m.ph  ==>
        |         |         |-%s.zz  ==>
        |         |         |-h2  ==>
        |         |
        |         |-toc-------EMPTY
        |         |-figlist---EMPTY
        |
        |-body------h0--------h0t-------%m.ph  ==>
        |         |         |-%s.zz  ==>
        |         |         |-h1--------h1t-------%m.ph  ==>
        |         |                   |-%s.zz  ==>
        |         |                   |-h2  ==>
        |         |
        |         |-h1--------h1t-------%m.ph  ==>
        |                   |-%s.zz  ==>
        |                   |-h2  ==>
        |
        |-appendix-h1--------h1t-------%m.ph  ==>
        |                   |-%s.zz  ==>
        |                   |-h2  ==>
        |
        |-backm-----%bm.d----glossary--h1t-------%m.ph  ==>
        |         |         |         |-%s.zz  ==>
        |         |         |         |-h2  ==>
        |         |         |
        |         |         |-bibliog--h1t-------%m.ph  ==>
        |         |                   |-%s.zz  ==>
        |         |                   |-h2  ==>
        |         |
        |         |-h1--------h1t-------%m.ph  ==>
        |         |         |-%s.zz  ==>
        |         |         |-h2  ==>
        |         |
        |         |-index-----EMPTY

        +ix
        +%i.float
```

Figure 3 — Overall structure of the basic document

A front matter element **(frontm)** consists of a title page **(titlep)**, followed by an abstract **(abstract)**, or a preface **(preface)**, or headed section one elements **(h1)** (optional and repeatable), followed by a table of contents **(toc)** (optional), followed by a list of figures **(figlist)** (optional). The end-tag of the front matter can be omitted.

The title page element **(titlep)** consists of a title **(title)**, and a document number **(docnum)** (optional), and a date **(date)** (optional), and an abstract **(abstract)** (optional), and author **(author)**, or address **(address)**, or "section subelements" **(%s.zz)** (optional and repeatable). All these may be entered in any order. The end-tag of the title page can be omitted.

A title element **(title)** consists of one or more title lines **(tline)**, each in its turn consisting of "phrase models" **(%m.ph)**. The end-tag of the title element can be omitted. The end-tag of the title lines can be omitted (and the start-tag of the first title line).

A document number **(docnum)**, or a date **(date)**, or an author element **(author)** consists of character data **(#PCDATA)**. The end-tag of these elements can be omitted.

An address element **(address)** consists of one or more address lines **(aline)**, each in its turn consisting of "phrase models" **(%m.ph)**. No tag of the address element can be omitted. The end-tag of the address lines can be omitted (and the start-tag on the first address line).

An abstract **(abstract)**, or a preface element **(preface)** consists of a title **(h1t)**, followed by "section subelements" **(%s.zz)** (optional and repeatable), followed by headed section two elements **(h2)** (optional and repeatable). The end-tag of the abstract or preface element can be omitted.

A table of contents **(toc)**, or a list of figures element **(figlist)** has no content; the content is generated by the application software based on information contained in other elements. The end-tag of these elements can be omitted.

A body element **(body)** consists of either one or more headed section zero elements **(h0)**, or one or more headed section one elements **(h1)**. The end-tag of the body element can be omitted.

A headed section zero element **(h0)** consists of a title **(h0t)**, followed by "section subelements" **(%s.zz)** (optional and repeatable), followed by one or more headed section one elements **(h1)**.

A headed section one element **(h1)** consists of a title **(h1t)**, followed by "section subelements" **(%s.zz)** (optional and repeatable), followed by headed section two elements **(h2)** (optional and repeatable). The end-tag can be omitted.

```
h2————————h2t————————%m.ph   ==>
          |
          |–%s.zz   ==>
          |
          |–h3————————h3t————————%m.ph   ==>
                      |
                      |–%s.zz   ==>
                      |
                      |–h4————————h4t————————%m.ph   ==>
                                  |
                                  |–%s.zz   ==>
```

Figure 4 — Structure of the h2 element

A headed section two element **(h2)** consists of a title **(h2t)**, followed by "section subelements" **(%s.zz)** (optional and repeatable), followed by headed section three elements **(h3)** (optional and repeatable). The end-tag can be omitted.

A headed section three element **(h3)** consists of a title **(h3t)**, followed by "section subelements" **(%s.zz)** (optional and repeatable), followed by headed section four elements **(h4)** (optional and repeatable). The end-tag can be omitted.

A headed section four element **(h4)** consists of a title **(h4t)**, followed by "section subelements" **(%s.zz)** (optional and repeatable). The end-tag can be omitted.

The title elements of headed sections **(h0t - h4t)** consist of "phrase models" **(%m.ph)**. Both the start and end-tags can be omitted.

An appendix element **(appendix)** consists of one or more headed section one elements **(h1)**.

A back matter element **(backm)** consists of a glossary **(glossary)**, or a bibliography **(bibliog)**, or a headed section one element **(h1)** (optional and repeatable), followed by an index **(index)** (optional). The end-tag can be omitted.

A glossary **(glossary)**, or a bibliography element **(bibliog)** consists of a title **(h1t)**, followed by "section subelements" **(%s.zz)** (optional and repeatable), followed by headed section two elements **(h2)** (optional and repeatable). The end-tag of these elements can be omitted.

An index element **(index)** has no content; the content is generated by the application software based on information contained in other elements. The end-tag can be omitted.

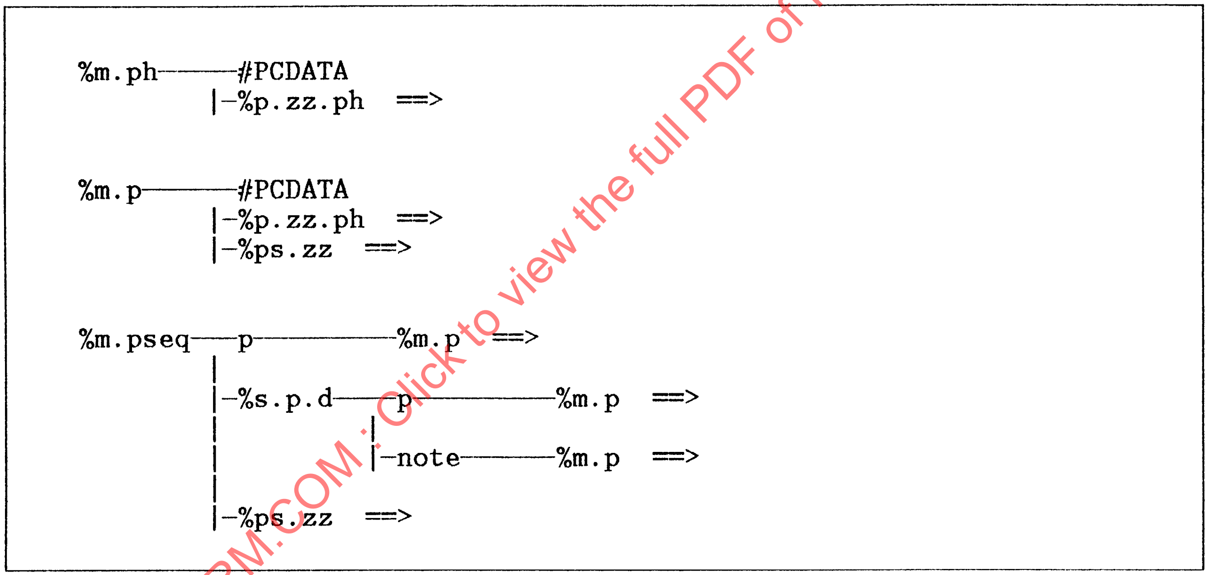


Figure 5 — Structure of the phrase (%m.ph), paragraph (%m.p), and paragraph sequence (%m.pseq) models

A "phrase model" **(%m.ph)** is defined as character data **(#PCDATA)** or "phrases" **(%p.zz.ph)** (optional and repeatable).

A "paragraph model" **(%m.p)** is defined as character data **(#PCDATA)**, or "phrases" **(%p.zz.ph)**, or "paragraph/section subelements" **(%ps.zz)** (optional and repeatable).

A "paragraph sequence" **(%m.pseq)** is defined as a paragraph **(p)**, followed by a "simple paragraph" **(%s.p.d)** or "paragraph/section subelements" **(%ps.zz)** (optional and repeatable).

A "simple paragraph" **(%s.p.d)** is defined as a paragraph **(p)** or a note **(note)**.

A paragraph **(p)** or a note element **(note)** consists of "paragraph models" **(%m.p)**. Both the start and end-tag of these elements can be omitted.

NOTE — The start-tag can be omitted only under certain, very restricted, conditions.

```
%s.zz———%s.p.d———p——————————%m.p  ==>
         |         |
         |         |-note————%m.p  ==>
         |
         |-%ps.zz  ==>
         |
         |-%s.top———top1—————%m.top———th————————%m.ph  ==>
                   |                   |
                   |                   |-p—————————%m.p  ==>
                   |                   |
                   |                   |-%s.zz——————...
                   |         -top1
                   |
                   |-top2—————%m.top———th————————%m.ph  ==>
                   |                   |
                   |                   |-p—————————%m.p  ==>
                   |                   |
                   |                   |-%s.zz——————...
                   |         -top2
                   |
                   |-top3—————%m.top———th————————%m.ph  ==>
                   |                   |
                   |                   |-p—————————%m.p  ==>
                   |                   |
                   |                   |-%s.zz——————...
                   |         -top3
                   |
                   |-top4—————%m.top———th————————%m.ph  ==>
                                       |
                                       |-p—————————%m.p  ==>
                                       |
                                       |-%s.zz——————...
                             -top4
```

Figure 6 — Structure of the section subelements

A "section subelement" **(%s.zz)** is defined as a "simple paragraph" **(%s.p.d)**, or a "paragraph/section subelement" **(%ps.zz)**, or a "topic" **(%s.top)**.

A topic of level one **(top1)** consists of a topic heading **(th)** (optional), followed by a paragraph **(p)**, followed by "section subelements" **(%s.zz)** (optional and repeatable). Topics of level one are excluded elements; thus they cannot occur further at any level of this topic level. The end-tag can be omitted.

A topic heading consists of a "phrase model" **(%m.ph)**. The end-tag can be omitted.

A topic of level two **(top2)** consists of a topic heading **(th)** (optional), followed by a paragraph **(p)**, followed by "section subelements" **(%s.zz)** (optional and repeatable). Topics of level two are excluded elements; thus they cannot occur further at any level of this topic level. The end-tag can be omitted.

A topic of level three **(top3)** consists of a topic heading **(th)** (optional), followed by a paragraph **(p)**, followed by "section subelements" **(%s.zz)** (optional and repeatable). Topics of level three are excluded elements; thus they cannot occur further at any level of this topic level. The end-tag can be omitted.

A topic of level four **(top4)** consists of a topic heading **(th)** (optional), followed by a paragraph **(p)**, followed by "section subelements" **(%s.zz)** (optional and repeatable). Topics of level four are excluded elements; thus they cannot occur further at any level of this topic level. The end-tag can be omitted.

```
%p.zz.ph—%p.em.ph—hp1————————%m.ph  ==>
      |          |
      |          |-hp2————————%m.ph  ==>
      |          |
      |          |-hp3————————%m.ph  ==>
      |          |
      |          |-hp0————————%m.ph  ==>
      |          |
      |          |-cit————————%m.ph  ==>
      |
      |-%p.rf.ph—hdref————————%m.ph  ==>
      |          |
      |          |-figref—————%m.ph  ==>
      |
      |-%p.rf.d——fnref————————EMPTY
      |          |
      |          |-liref——————EMPTY
      |
      |-q————————————%m.ph  ==>
```

Figure 7 — Structure of all phrases

A "phrase" **(%p.zz.ph)** is defined as an "emphasized phrase" **(%p.em.ph)**, or a "reference phrase" **(%p.rf.ph)**, or a "reference (empty)" **(%p.rf.d)**, or a "quotation" **(q)**.

An "emphasized phrase" **(%p.em.ph)** is defined as a highlighted phrase of level zero to three **(hp0 - hp3)**, or a citation **(cit)**.

A highlighted phrase of level zero to three **(hp0 - hp3)**, or a citation **(cit)** consists of "phrase models" **(%m.ph)**. No markup minimization is permitted.

NOTE — As a "phrase model" contains character data, or "phrases" the definition is recursive, which implies that highlighted phrases can be nested.

A "reference phrase" **(%p.rf.ph)** is defined as a header reference **(hdref)**, or a figure reference **(figref)**.

A header reference element **(hdref)**, or a figure reference **(figref)** consists of "phrase models" **(%m.ph)**. No markup minimization is permitted.

A "reference (empty)" **(%p.rf.d)** is defined as a footnote reference **(fnref)**, or a list item reference **(liref)**.

A footnote reference element **(fnref)**, or list item reference **(liref)** has no content; the content is generated by the application software.

A (short) quotation element **(q)** consists of a “phrase model” **(%m.ph)**. No markup minimization is permitted.

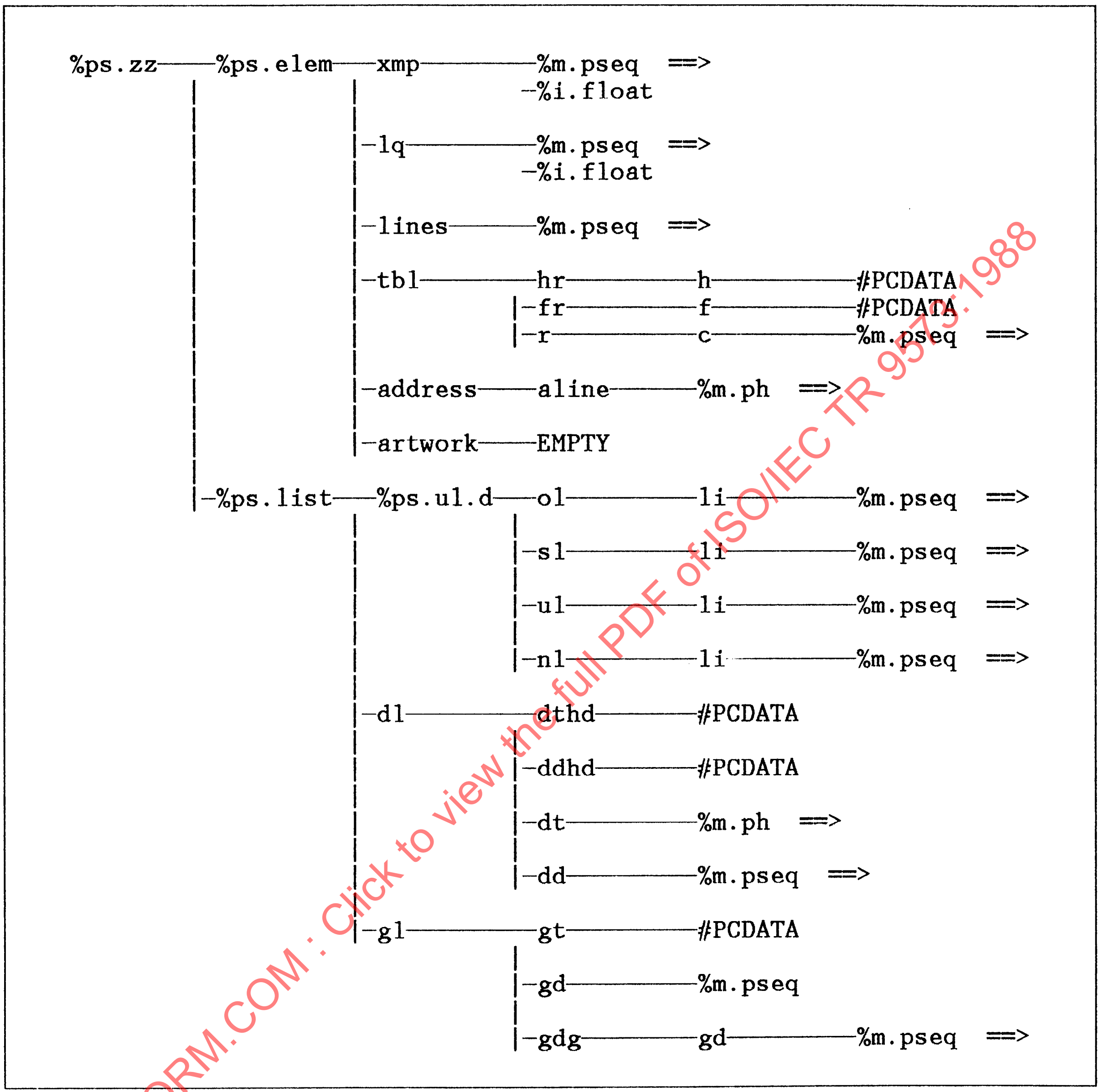


Figure 8 — Structure of all paragraph/section subelements

A “paragraph/section subelement” **(%ps.zz)** is defined as “other elements” **(%ps.elem)** or “lists” **(%ps.list)**.

“Other elements” **(%ps.elem)** is defined as example **(xmp)**, or long quotation **(lq)**, or lines **(lines)** or table **(tbl)**, or address **(address)**, or artwork **(artwork)**.

An example **(xmp)**, or a long quotation **(lq)** consists of a “paragraph sequence” **(%m.pseq)**. “Floating elements” **(%i.float)** are excluded elements; thus they cannot occur further at any level in an example or a long quotation. No markup minimization is permitted.

A lines element **(lines)** consists of a “paragraph sequence” **(%m.pseq)**. Both the start-tag and the end-tag can be omitted.

A table element **(tbl)** consists of a heading row **(hr)** (optional and repeatable), followed by a footing row **(fr)** (optional and repeatable), followed by one or more rows **(r)**. No markup minimization is permitted.

A heading row **(hr)** consists of one or more heading row cells **(h)**. The end-tag can be omitted.

A footing row **(fr)** consists of one or more footing row cells **(f)**. The end-tag can be omitted.

A heading row cell **(h)** or footing row cell **(f)** consists of character data **(#PCDATA)**. Both the start and end-tag can be omitted.

A row **(r)** consists of one or more cells **(c)**. Both the start and end-tags can be omitted.

A cell **(c)** consists of a "paragraph sequence" **(%m.pseq)**. Both the start and end-tags can be omitted.

An artwork element **(artwork)** has no content. Using attributes, space is reserved for pasted-in material.

"Lists" **(%ps.list)** are defined as "unit item lists" **(%ps.ul.d)**, or definition list **(dl)**, or glossary list **(gl)**.

"Unit item lists" **(%ps.ul.d)** are defined as ordered list **(ol)**, or simple list **(sl)**, or unordered list **(ul)**, or note list **(nl)**. No markup minimization is permitted.

Ordered list **(ol)**, or simple list **(sl)**, or unordered list **(ul)**, or note list elements **(nl)** consists of list items **(li)** (optional and repeatable).

List items **(li)** consist of "paragraph sequences" **(%m.pseq)**. The end-tag can be omitted.

A definition list **(dl)** consists of optional headings for the term **(dthd)** and definition **(ddhd)**, followed by optional and repeatable blocks of definition term **(dt)** (repeatable) and definition **(dd)**. No markup minimization is permitted.

A definition heading **(dthd, ddhd)** consists of character data **(#PCDATA)**. The end-tag can be omitted.

A definition term **(dt)** consists of a "phrase model" **(%m.ph)**. The end-tag can be omitted.

A definition **(dd)** consists of a "paragraph sequence" **(%m.pseq)**. The end-tag can be omitted.

A glossary list **(gl)** consists of an optional and repeatable block of a glossary term **(gt)**, followed by a glossary definition **(gd)** or a glossary definition group **(gdg)**. No markup minimization is permitted.

A glossary term **(gt)** consists of character data **(#PCDATA)**. The end-tag can be omitted.

A glossary definition **(gd)** consists of a "paragraph sequence" **(%m.pseq)**. The end-tag can be omitted.

A glossary definition group **(gdg)** consists of one or more glossary definitions **(gd)**. The end-tag can be omitted.

"Floating elements" **(%i.float)** are defined as figures **(fig)** and footnotes **(fn)**.

A figure **(fig)** consists of a figure body **(figbody)**, followed by an optional block of figure caption **(figcap)**, followed by an optional figure description **(figdesc)**. "Floating elements" are excluded elements; thus a figure or a footnote cannot occur inside a figure. No markup minimization is permitted.

```
%i.float—fig——————————figbody———%m.pseq  ==>
        |           |
        |           |-figcap————%m.ph  ==>
        |           |
        |           |-figdesc———%m.pseq  ==>
        |
        | -%i.float
        |
        |-fn——————————%m.pseq  ==>

          -%i.float
```

Figure 9 — Structure of all floating elements

A figure body **(figbody)** consists of a "paragraph sequence" **(%m.pseq)**. Both the start and end-tags can be omitted.

A figure caption **(figcap)** consists of a "phrase model" **(%m.ph)**. The end-tag can be omitted.

A figure description **(figdesc)** consists of a "paragraph sequence" **(%m.pseq)**. The end-tag can be omitted.

A footnote **(fn)** consists of a "paragraph sequence" **(%m.pseq)**. "Floating elements" are excluded elements; thus a figure or a footnote cannot occur inside a footnote. No markup minimization is permitted.

An index entry **(ix)** consists of character data **(#PCDATA)**. The end-tag can be omitted.

5.3 Description of the tags

5.3.1 Major document elements

general document <general> </general> security= status= version=

This element is at the top of the hierarchy called general document. "General" is therefore both a document type and an element type. A general document may consist of front matter, a main body, an appendix section, and back matter, in that order, only the main body being necessary. The tag <general> is the first thing keyed in before any text. </general>, the end-tag, being the last tag of the document, is keyed in, after all the text has been input. An attribute, if present, is keyed in after the generic identifier and before the tag-close character. The security attribute (security=) may be used to indicate the confidential nature of a document, where an implementation may cause the value of this attribute to appear on every page, or just the front cover of the document. It could also be used by a system to restrict access to the document itself. The status attribute (status=) may be used to indicate that the document is at a draft stage, again where the implementation may cause the value of this attribute to appear on every page, or just the front cover of the document. The version attribute (version=) may be used to indicate a version number of a document. <general> must be followed immediately by another tag.

```
<general sec="Company Confidential" status="Draft">
```

front matter <frontm>

If there is front matter, the second tag input is <frontm>, where this may consist of information for a title page, and other elements such as an abstract. The elements may be in any order. No end-tag is necessary, the completion of front matter being implied by the beginning of the main body. <frontm> is followed immediately by another tag.

main body <body>

This tag precedes the elements comprising the main body of the document. No end-tag is necessary. <body> must be present and is followed immediately by another tag, indicating the start of an element contained in the main body.

appendix section <appendix>

One or more appendices to the document are preceded by <appendix>. No end-tag is necessary. <appendix> is followed immediately by the tag for a headed section one element, <h1>.

back matter <backm>

Any glossary or bibliography for the publication as a whole is placed at the end, and an index if required. The tag <backm> denotes that there is back matter. No end-tag is necessary. It is followed immediately by another tag.

table of contents <toc>

If a table of contents is required, key in <toc>. No text for the list is input, the contents list being generated by an application program from headings, adding page numbers. <toc> is followed immediately by another tag. Valid only in the front matter.

figure list <figlist>

When a list of figures is required, key in <figlist>. No text for the list is input, the contents for a list of figures being created from figure captions (and associated page numbers). <figlist> is followed immediately by another tag. Valid only in the front matter.

index <index>

The presence of the index tag <index> indicates that an index is to be included in the back matter at the point indicated by <index>. Its constituent elements are tagged throughout the document, as appropriate. It is followed immediately by another tag.

5.3.2 Title page elements

title page <titlep>

<titlep> indicates the beginning of information to appear on the title page(s) of the document. The title page information may consist of, keyed in any order, a title, an optional document number, an optional date, an optional abstract, optional author information, and optional section subelements. <titlep> is followed immediately by another tag. Valid only in the front matter element.

title (of document) <title> stitle=

This tag precedes the title of the document. The short title attribute (stitle=) may be used to input a short version of the title to be used, typically, as a running head or foot. No end-tag is necessary. Valid only in the title page element. The content of the title element is one or more title lines (tline elements). The start and end-tag of these can be omitted.

```
<title stitle="Good SGML Practice">
The Handbook of
Good SGML Practice
```

document number <docnum>

If a document is to be given a number or code, it is preceded by <docnum>. No end-tag is necessary. Valid only in the title page element.

```
<docnum>ISO 8879
```

date <date>

This is the date associated with the publication of the document. Whereas in draft stages it could indicate the day and the month as well as the year, the final version could merely indicate the year. In draft stages the date could be system-supplied. No end-tag is necessary.

```
<date>1987-06-29
```

author <author>

The name of the author is preceded by <author>. The element may be repeated in the case of there being more than one author. No end-tag is necessary. Valid only in the title page element.

```
<author>A.Someone
<author>B.Else
```

address <address> </address>

The <address> tag indicates the start of an address. It is followed immediately by repeatable address line (aline) elements, where the end-tags can be omitted. When all the lines of the address have been keyed in, </address> is input to indicate the end of the address. The element may be repeated in the case of there being more than one address. Valid only in the title page element.

```
<address>
<aline>1, rue de Varembé
<aline>CH-1211 Genève 20
<aline>Suisse
</address>
```

5.3.3 Front matter elements

abstract <abstract> id= stitle=

The abstract or summary of a document is preceded by <abstract>. It is followed, after a required title, by a tag for one of the "section subelements". The value of the identifier attribute (id=) can be used when referring to the abstract and must be a unique reference in the document. The short title attribute (stitle=) may be used to input a short version of the heading to be used, typically, as a running head or foot. Valid only in the front matter.

```
<abstract>Abstract
<p>This report aims to show examples of good usage of SGML in
a variety of application areas.
```

preface <preface> id= stitle=

The <preface> tag is keyed in before a preface. The element may be repeated in the case of there being more than one preface. It is followed, after a required title, by a tag for one of the "section subelements". The value of the identifier attribute (id=) can be used when referring to the preface heading and must be a unique reference in the document. The short title attribute (stitle=) may be used to input a short version of the heading to be used, typically, as a running head or foot. No end-tag is necessary. Valid only in the front matter.

```
<preface>Introduction
<p>ISO 8879 contains, without much explanation, an example of
a document type definition ...
```

5.3.4 Back matter elements

glossary <glossary> id= stitle=

The glossary is preceded by <glossary>. It is followed, after a required title, by a tag for one of the "section subelements". The value of the identifier attribute (id=) can be used when referring to the glossary and must be a unique reference in the document. The short title attribute (stitle=) may be used to input a short version of the heading to be used, typically, as a running head or foot. No end-tag is necessary. Valid only in the back matter.

```
<glossary>Glossary
<gl>
<gt>Beaujolais Nouveau<gd>A wine from the Beaujolais district
bottled and distributed very young.
</gl>
```

bibliography <bibliog> id= stitle=

The <bibliog> tag precedes any list of bibliographic details to be input. It is followed, after a required title, by a tag for one of the "section subelements". The value of the identifier attribute (id=) can be used when referring to the bibliography and must be a unique reference in the document. The short title attribute (stitle=) may be used to input a short version of the heading to be used, typically, as a running head or foot. Valid only in the back matter.

5.3.5 Headed sections

headed section 0-4 <h0>, <h1>, <h2> id= stitle=
<h3>, <h4> id=

These tags are used to identify the main sections of the document. Level 0 could identify parts of a large document, level 1 chapter titles, the others being used for the subsections. The value of the identifier attribute (id=) can be used when referring to the heading (section) and must be a unique reference in the document. The short title attribute (stitle=) may be used to input a short version of the heading to be used, typically, as a running head or foot. No end-tag is necessary.

```
<h1 id=BAS stitle="Becoming Acquainted with SGML">
Becoming Acquainted with the Standard Generalized Markup
Language
```

5.3.6 Topics

topics of level 1-4 <top1>, <top2>, <top3>, <top4>

These tags are used to identify topic-subdivisions in the document. They can be followed by an optional topic heading tag. The end-tag can be omitted.

NOTE — Topics differ from headed sections in that headed sections must be used in strict head level order: e.g. <H1>Text <H2>Text... For topics, however, there are no strict levels: e.g. <top3>Text <top1>Text <top2>Text </top2></top1></top3> is permitted. It is, however, not permitted to imbed one topic of a particular level within another topic of the same level.

topic heading < th>

The <th> tag is used to identify a topic heading. It immediately follows one of the topic tags. The end-tag can be omitted.

```
<top1><th>Important considerations
<p>When using SGML it is important to ...
```

5.3.7 Tabular material

tabular material <tbl> </tbl> cols=

The table tag <tbl> precedes tabular material, an end-tag being required. The required columns attribute (cols=) specifies the number of columns in the table. <tbl> is followed immediately by another tag. The end-tag </tbl> is input after the table elements for this table.

heading row < hr>

This tag is used to identify heading rows in a table. It is followed by text for the first cell of the heading. The end-tag can be omitted.

heading in heading row < h>

This tag is used to identify the headings for each individual cell in a table heading row. The end-tag can be omitted.

footing row < fr>

This tag is used to identify footing rows in a table. It is followed by text for the first cell of the footing. The end-tag can be omitted.

heading in footing row < f>

This tag is used to identify the headings for each individual cell in a table footing row. The end-tag can be omitted.

row in body of table < r>

This tag is used to identify each row in the body of a table. Both the start and end-tags can be omitted, in certain situations.

cell in a row <c> heading=

This tag is used to identify each cell in a row in the body of a table. If heading=h attribute is specified the cell is a row heading. Both the start and end-tags can be omitted.

```
<tbl>
<hr>Country <h>Language <h>Currency
<r>Sweden   <c>Swedish  <c>Krona
<r>France   <c>French   <c>Franc
</tbl>

<tbl>
<hr>Restaurants with stars <h>France<h>Netherlands<h>Belgium
<r><c heading=h>one star    <c> 549  <c>    37      <c>  55
<r><c heading=h>two stars   <c>  58  <c>     5      <c>  15
<r><c heading=h>three stars<c>  17  <c>     0      <c>   2
</tbl>
```

5.3.8 Figure

figure <fig> </fig> id= frame= place= width= align= lines=

The figure tag <fig> precedes figure text, a diagram or other illustration, an end-tag being required. The value of the identifier attribute (id=) can be used when referring to the figure and must be unique in the document. The frame attribute (frame=) takes one of the values: box, rule, or none. The value "box" means that the figure should be enclosed in a box, rule means that a horizontal rule will be used before and after the figure, and none means that no frame will be used. None is the default. The place attribute (place=) takes one of the values: top, fixed, or bottom. Top means that the figure will be floated to the top of the next page, bottom that it will be floated to the bottom of the current page, if there is room for it, or to the bottom of the next page if not; the remainder of the current output column will be filled with the text after the </fig> tag. Fixed means that if the figure will not fit the current output column, the rest of the column will be left blank and the figure will be positioned at the top of the next column. The width attribute (width=) takes the value column or page for a column-wide or page-wide figure. The align attribute (align=) specifies how the figure should be aligned horizontally in the column or on the page. It takes the value left, right, or center, center being the default. The lines attribute (lines=) specifies whether text in the figure body should be left "as is", not formatted, (lines=lines), or if normal flowing text should be used (lines=flow). Associated with a figure may be a figure caption and a figure description. The end-tag </fig> is input after all the figure elements.

```
<fig id=DR>
<figcap>Distribution of Rainfall
<figdesc>The amount of rainfall is influenced by the position of
the Pennines and the prevailing wind conditions.
</fig>
```

figure body < figbody >

This element, where both the start and end-tags can be omitted, immediately follows the <fig> tag.

figure caption < figcap >

The figure caption is preceded by <figcap>. No end-tag is necessary. Valid only in a figure element.

figure description < figdesc >

The description of a figure is preceded by <figdesc>. No end-tag is necessary. Valid only in a figure element.

artwork <artwork> sizex= sizey=

This tag is used to reserve space for pasted in artwork. Normally this tag would be used in a figure body, but it can also be used elsewhere. Attributes are used to specify the size of the artwork; the height is specified with the required sizey attribute. The sizex attribute is optional with the default being the current text width in the column. The artwork element has no content and the end-tag can be omitted.

```
<artwork sizey=110 sizex=134>
```

5.3.9 Phrases

highlighted phrases 0-3 <hp1>, <hp2>, <hp3>, <hp0>
</hp1>, </hp2>, </hp3>, </hp0>

These tags variously precede a word or a phrase to be highlighted, where there are different kinds of highlighting. The four kinds could be: 1 italics; 2 bold; 3 bold italics; 0 no emphasis. An end-tag is required. Highlighted phrases may be nested.

```
The term <hp1>element</hp1> has a special meaning in the
context of SGML.
<hp3>This long highlighted phrase has <hp0>one</hp0> non-highlighted
word</hp3>.
```

citations <cit> </cit>

References to other works are identified using the <cit> tag. The end-tag is required.

```
The standard <cit>ISO 8879</cit> is useful for SGML implementors.
```

short quotation <q> </q>

A short quotation, occurring within running text, to be enclosed within quotation marks, is preceded by <q> and followed by </q>. (This distinguishes it from a word or phrase that would appear within quotation marks for a different purpose.) On output it would normally appear within curved quotation marks, double or single, dependent on the house style or the language used. Quotations may be nested.

```
The report states clearly <q>the tolerances shall not
exceed the standard values</q>.
```

long quotation <lq> </lq>

Longer quotations are usually set in a different style from short ones, and are therefore treated differently, being delimited by <lq> and </lq>.

5.3.10 Lists

The "general" document includes six types of lists:

— "ordered" list, where there may be a need to refer to each item in the list. Typically the list items are numberred by the text-formatter.

— "unordered" list, where there is no need to refer to any item in the list, but where each item should be clearly standing out. Typically the list items are indicated by bullets, stars, or dashes by the text-formatter.

— "simple" list, where there is no need to refer to any item in the list, and no need for each item to be clearly standing out. Typically the list items are indicated by indentation by the text-formatter.

— "note" list — a list of notes. Typically the list items are numberred by the text-formatter.

— "definition" list where "terms" or "symbols" are defined.

— "glossary" list where "words" or "phrases" are explained.

definition list <dl> </dl> compact= headhi= termhi= tsize=

The <dl> tag precedes a definition list. After the input for the definitions has been completed the end-tag </dl> is added. <dl> is followed immediately by either the tag for a definition term, <dt>, or the tag for the optional headings. The compact attribute (compact=) is used to indicate spacing within the list; compact=compact could mean that there is no extra white space between different items but extra white space is added before and after the list. If this attribute is omitted, it could mean the inclusion of white space both before and after the list as well as between the entries. The heading and term highlight attributes (headhi=, and termhi=) indicate the highlight level to be used for these elements (see the description of the <hp1> — <hp3> tags). The default highlight value is 2. The term-size attribute (tsize=) is used to specify the indent to be used for the definition. It is normally larger than the maximum width of the terms. The default value is 9 (en spaces).

```
<dl>
<dthd>Acronym <ddhd>Meaning
<dt>DTD       <dd>Document Type Definition
<dt>SGML      <dd>Standard Generalized Markup Language
</dl>
```

definition term heading <dthd>

If headings for the definition terms are present, this tag is used to identify them. The end-tag can be omitted. Valid only in the definition list element.

definition definition heading
<ddhd>

This tag is used to identify the heading for the definition of a term, if headings are present. Valid only in the definition list element.

definition term <dt>

This tag precedes the word or phrase being defined. No end-tag is required. It is followed by either another <dt> tag or the tag for the definition. Valid only in the definition list element.

definition definition <dd>

This tag precedes the definition of the term. The end-tag can be omitted. Valid only in the definition list element.

glossary list <gl> </gl> compact= termhi=

The <gl> tag precedes a glossary list. When the input for the glossary has been completed, key in </gl>. <gl> is followed immediately by the tag for a glossary term, <gt>. The compact attribute (compact=) is used to indicate spacing within the list; compact=compact could mean that there is no extra white space between different items but extra white space is added before and after the list. If this attribute is omitted, it could mean the inclusion of white space both before and after the list as well as between the entries. The term highlight attribute (termhi=) indicates the highlight level to be used for this element (see the description of the <hp1> — <hp3> tags). The default highlight value is 2.

```
<gl>
<gt>attribute
<gd>A string for providing supplementary information about
an element, that qualifies the generic identifier.
<gt>document type definition
<gd>...
 ...
</gl>
```

glossary term <gt>

This tag precedes the word or phrase being defined. No end-tag is necessary. It is followed immediately by the definition, <gd> or <gdg>. Valid only in the glossary list element.

glossary definition group <gdg>

For cases where there is more than one definition per term, the <gdg> tag is used to indicate the grouping of these elements. The end-tag can be omitted. Valid only in the glossary list element.

```
<gl>
<gt>grivna
<gdg><gd>Russian weight unit, 409 grams, used in the
15th to 19th centuries.
<gd>Russian term for the 10 kopek coin, also called grivennik,
first made in 1701.
</gl>
```

glossary definition <gd> source=

The definition of the associated term is preceded by <gd>. No end-tag is necessary. Valid only in the glossary list element. The source attribute (source=) is character data indicating the source of the definition. It could be presented as a "footnote" or in smaller typeface, or small caps, after the definition.

ordered list <ol> </ol> compact=

The <ol> tag precedes an ordered list. The compact attribute (compact=) is used to indicate spacing within the list; compact=compact could mean that there is no extra white space between different items but extra white space is added before and after the list. If this attribute is omitted, it could mean the inclusion of white space both before and after the list as well as between the entries. When all the individual items have been input, the end-tag </ol> is required. Lists may be nested. <ol> is followed by <li>.

simple list <sl> </sl> compact=

The <sl> tag precedes a simple list. The compact attribute (compact=) is used to indicate spacing within the list; compact=compact could mean that there is no extra white space between different items but extra white space is added before and after the list. If this attribute is omitted, it could mean the inclusion of white space both before and after the list as well as between the entries. When all the individual items have been input, the end-tag </sl> is required. Lists may be nested. <sl> is followed by <li>.

unordered list <ul> </ul> compact=

The <ul> tag precedes an unordered list. The compact attribute (compact=) is used to indicate spacing within the list; compact=compact could mean that there is no extra white space between different items but extra white space is added before and after the list. If this attribute is omitted, it could mean the inclusion of white space both before and after the list as well as between the entries. When all the individual items have been input, the end-tag </ul> is required. Lists may be nested. <ul> is followed immediately by <li>.

note list <nl> </nl> compact=

The <nl> tag precedes a list of notes. The compact attribute (compact=) is used to indicate spacing within the list; compact=compact could mean that there is no extra white space between different items but extra white space is added before and after the list. If this attribute is omitted, it could mean the inclusion of white space both before and after the list as well as between the entries. When all the individual items have been input, the end-tag </nl> is required. Lists may be nested. <nl> is followed by <li>.

list item <li> id=

Each individual list item is preceded by <li>. The value of the identifier attribute (id=) can be used when referring to the item and must be a unique reference in the document. No end-tag is necessary. Valid only in a list element.

5.3.11 References

heading reference <hdref> refid= page=

The heading reference is used within the document to refer to a heading. If a value for the reference identifier attribute (refid=) is given the reference is provided by the application, the content of the element is empty and no end-tag is required. The style of the reference would typically be the heading number or the heading title. The page number attribute (page=) can take the values yes and no; in the first case, which is the default, the appropriate page number, supplied by the system, is added to the reference. Alternatively, the refid attribute can be omitted and the reference text included in the content of the element. In this case the end-tag is required.

NOTE — It is recommended that the latter option be used only in cases of references to headings external to the document being processed.

```
See <hdref refid=BAS page=no>.
See <hdref>Chapter 3 in Part IV</hdref>.
```

figure reference <figref> refid= page=

The figure reference is used within the document to refer to a figure. If a value for the reference identifier attribute (refid=) is given the reference is provided by the application, the content of the element is empty and no end-tag is required. The style of the reference would typically be the figure number and, perhaps, a caption. The page number attribute (page=) can take the values yes and no; in the first case, which is the default, the appropriate page number, supplied by the system, is added to the reference. Alternatively, the refid attribute can be omitted and the reference text included in the content of the element. In this case the end-tag is required.

NOTE — It is recommended that the latter option be used only in cases of references to figures external to the document being processed.

```
Reference should be made to <figref refid=DR>.
Reference should be made to <figref>Figure 7 in Part IV</figref>.
```

footnote reference <fnref> refid=

The footnote reference is used within the document to refer to the identified footnote. The appropriate value for the reference identifier attribute (refid=) must be provided. The tag is self-contained.

```
One use of a footnote in ISO documents is given in
<fnref refid=NF>.
```

list item reference <liref> refid= page=

The list item reference is used within the document to refer to an identified list item by which it is replaced. The page number attribute (page=) can take the values yes and no; in the first case, which is the default, the appropriate page number, supplied by the system, is added to the reference. The appropriate value for a reference identifier attribute (refid=) must be provided. The tag is self-contained.

5.3.12 Basic elements

paragraph <p>

Each paragraph is preceded by <p>. No end-tag is necessary.

note <note>

A note to be included in the text is preceded by <note>. This would be treated in a different way from a footnote. (Many documents have both notes printed at the point of occurrence, and footnotes printed at the bottom of the page in which reference is made to them.) When all the text or paragraphs of the note have been keyed in, </note> is input to indicate the end of the note.

```
<note>It is extremely important to be consistent in the
design of a document type definition.</note>
```

example <xmp> </xmp> depth= keep= lines=

The example element differs from that of a figure in that it has no caption, nor can it be referred to from other parts of the document. It could be just one line. The end-tag </xmp> is input after the example. The depth attribute (depth=) can be used to reserve space for a pasted in example. The keep attribute (keep=) is used to specify whether an example can be split or not if it does not fit in the remainder of the current column. The default is all, indicating no split permitted. A numeric value would indicate the minimum number of lines to be kept together. The lines attribute (lines=) specifies whether text in the example should be left "as is", not formatted, (lines=lines), or if normal flowing text should be used (lines=flow).

```
<xmp>When you press the enter key
something will be displayed on the screen.
</xmp>
```

which could result in

```
When you press the enter key
something will be displayed on the screen.
```

```
<xmp lines=lines>First record.
Second record.
</xmp>
```

which could result in

```
First record.
Second record.
```

```
<xmp lines=flow>First record.
Second record.
</xmp>
```

which could result in

```
First record.  Second record.
```

```
<xmp depth=5>
</xmp>
```

to reserve five lines for a pasted-in example.

footnote <fn> </fn> id=

The <fn> tag precedes text for a footnote that should be input at the point of occurrence. The application software will cause an indicator to be inserted at that point, and cause the footnote itself to appear at the bottom of the page. The value of the identifier attribute (id=) can be used when referring to the footnote and must be a unique reference in the document. The footnote text is terminated by </fn>.

```
Many publications contain both notes and footnotes.
<fn id=NF>The latter are used, for example, when
referring to a draft international standard to indicate
that publication is anticipated in due course.</fn>
Whereas notes are printed where they occur, footnotes are
collected at the bottom of the page in which reference is
made to them.
```

index entry <ix> id= print= see= seeid=

This tag precedes an entry to be placed in the index together with the current page number. The value of the identifier attribute (id=) can be used when referring to the index entry and must be a unique reference in the document. The print attribute (print=) specifies replacement text to be used instead of the index term following the <ix> tag. It is sorted, however, in the index according to the order of this term and not the replacement text. The term may be included in parentheses in the index. The see=word attribute is used to add a reference to another word in the index rather than to a particular page. The seeid attribute has a similar use as the see= attribute, but refers to the term associated with the index entry having the specified id.

```
<ix>Bourgogne
<ix see=Bourgogne>Clos de Vougeot
<ix id=BORD>Bordeaux
<ix seeid=BORD>Medoc
<ix>Claret
<ix print=Bordeaux>Claret
```

which could result in

```
Bordeaux . . . . . . . . . . . . . . 10
Claret (Bordeaux)  . . . . . . . . . 10
Bourgogne  . . . . . . . . . . . . . 15
Claret . . . . . . . . . . . . . . . 10
Clos de Vougeot (see Bourgogne)
Medoc (see Bordeaux)
```

5.4 Alphabetical list of the generic identifiers and their attributes

<abstract>	id= stitle=
<address> </address>	
<aline>	
<appendix>	
<artwork>	sizex= sizey=
<author>	
<backm>	
<bibliog>	id= stitle=
<body>	
<c>	heading=
<cit> </cit>	
<date>	
<dd>	
<ddhd>	
<dl> </dl>	compact= headhi= termhi= tsize=
<docnum>	
<dt>	
<dthd>	
<f>	
<fig> </fig>	id= frame= place= width= align= lines=
<figbody>	
<figcap>	
<figdesc>	
<figlist>	
<figref> </figref>	refid= page=
<fn> </fn>	id=
<fnref>	refid=
<fr>	
<frontm>	
<gd>	source=
<gdg>	
<general> </general>	security= status= version=
<gl> </gl>	compact= termhi=
<glossary>	id= stitle=
<gt>	
<h>	
<hdref> </hdref>	refid= page=
<hp0> </hp0>	
<hp1> </hp1>	
<hp2> </hp2>	
<hp3> </hp3>	
<hr>	
<h0>	id= stitle=
<h0t>	

<h1>	id= stitle=
<h1t>	
<h2>	id= stitle=
<h2t>	
<h3>	id=
<h3t>	
<h4>	id=
<h4t>	
<index>	
<ix>	id= print= see= seeid=
<li>	id=
<lines>	
<liref>	refid= page=
<lq> </lq>	
<nl> </nl>	compact=
<note>	
<ol> </ol>	compact=
<p>	
<preface>	id= stitle=
<q> </q>	
<r>	
<sl> </sl>	compact=
<tbl> </tbl>	cols=
<th>	
<title>	stitle=
<titlep>	
<tline>	
<toc>	
<top1>	
<top2>	
<top3>	
<top4>	
<ul> </ul>	compact=
<xmp> </xmp>	depth= keep= lines=

5.5 The document type definition

```
<!-- (C) International Organization for Standardization 1986
     Permission to copy in any form is granted for use with
     conforming SGML systems and applications as defined in
     ISO 8879, provided this notice is included in all copies.
-->
<!-- Public document type definition. Typical invocation:
<!DOCTYPE general PUBLIC "ISO 8879-1986//DTD General Document//EN" [
  <!ENTITY % ISOnum PUBLIC
    "ISO 8879-1986//ENTITIES Numeric and Special Graphic//EN">
  <!ENTITY % ISOpub PUBLIC
    "ISO 8879-1986//ENTITIES Publishing//EN">
  %ISOnum; %ISOpub;
      (Parameter entities and additional elements can be defined here.)
]>
-->
<!ENTITY % doctype "general" -- Document type generic identifier -->
<!--This is a document type definition for a "general" document.
It contains the necessary elements for use in many applications, and is
organized so that other elements can be added in the document type
declaration subset. -->

                  <!-- Entity Naming Conventions -->
<!--
                   Prefix = where used:
     p.   = in paragraphs (also in phrases if .ph suffix)
```

```
        s.   = in sections (i.e., among paragraphs)
        ps.  = in paragraphs and sections
        i.   = where allowed by inclusion exceptions
        m.   = content model or declared content
        a.   = attribute definition
        NONE= specific use defined in models
                    Suffix = allowed content:
        .ph  = elements whose content is %m.ph
        .d   = elements whose content has same definition
        NONE= elements with unique definitions
-->
                         <!-- Element Tokens -->
<!ENTITY % p.em.ph "hp1|hp2|hp3|hp0|cit" -- Emphasized phrases -->
<!ENTITY % p.rf.ph "hdref|figref" -- Reference phrases -->
<!ENTITY % p.rf.d  "fnref|liref" -- References (empty) -->
<!ENTITY % p.zz.ph "q|(%p.em.ph;)|(%p.rf.ph;)|(%p.rf.d;)" --All phrases-->
<!ENTITY % ps.ul.d "ol|sl|ul|nl" -- Unit-item lists -->
<!ENTITY % ps.list "%ps.ul.d;|dl|gl" -- All lists -->
<!ENTITY % ps.elem "xmp|lq|lines|tbl|address|artwork" -- Other elements
-->
<!ENTITY % ps.zz   "(%ps.elem;)|(%ps.list;)" -- Para/sect subelements -->
<!ENTITY % s.p.d   "p|note" -- Simple paragraphs -->
<!ENTITY % s.top   "top1|top2|top3|top4" -- Topics -->
<!ENTITY % s.zz    "(%s.p.d;)|(%ps.zz;)|(%s.top;)" -- Section subelements
-->
<!ENTITY % i.float "fig|fn" -- Floating elements -->
<!ENTITY % fm.d    "abstract|preface" --Front matter-->
<!ENTITY % bm.d    "glossary|bibliog" -- Back matter -->

                          <!-- Model Groups -->
<!ENTITY % m.ph    "(#PCDATA|(%p.zz.ph;))*" -- Phrase model -->
<!ENTITY % m.p     "(#PCDATA|(%p.zz.ph;)|(%ps.zz;))*" -- Paragraph model
-->
<!ENTITY % m.pseq  "(p, ((%s.p.d;)|(%ps.zz;))*)" -- Paragraph sequence -->
<!ENTITY % m.top   "(th?, p, (%s.zz;)*)" -- Topic model -->

                       <!-- Document Structure -->
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT %doctype;     - -  (frontm?, body, appendix?, backm?)
                               +(ix|%i.float;)>
<!ELEMENT frontm        - 0  (titlep, (%fm.d;|h1)*, toc?, figlist?)>
<!ELEMENT body          - 0  (h0+|h1+)>
<!ELEMENT appendix      - 0  (h1+)>
<!ELEMENT backm         - 0  ((%bm.d;|h1)*, index?)>
<!ELEMENT (toc|figlist|index)      -- Table of contents, figure list, --
                        - 0  EMPTY -- and index have generated content -->

                      <!-- Title Page Elements -->
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT titlep        - 0  (title & docnum? & date? & abstract? &
                                (author|address|%s.zz;)*)>
<!ELEMENT (docnum|date|author)
                        - 0  (#PCDATA) -- Document number, etc. -->
<!ELEMENT title         - 0  (tline+) -- Document title -->
<!ELEMENT tline         0 0  %m.ph; -- Title line -->

                        <!-- Headed Sections -->
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT h0            - 0  (h0t, (%s.zz;)*, h1+) -- Part -->
<!ELEMENT (h1|%bm.d;|%fm.d;)
                        - 0  (h1t, (%s.zz;)*, h2*) -- Chapter -->
<!ELEMENT h2            - 0  (h2t, (%s.zz;)*, h3*) -- Section -->
```

```
<!ELEMENT h3          - O  (h3t, (%s.zz;)*, h4*) -- Subsection -->
<!ELEMENT h4          - O  (h4t, (%s.zz;)*) -- Sub-subsection -->
<!ELEMENT  (h0t|h1t|h2t|h3t|h4t)
                      O O  %m.ph;    -- Headed section titles -->

                <!-- Topics (Captioned Subsections) -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT top1        - O  %m.top; -(top1) -- Topic 1 -->
<!ELEMENT top2        - O  %m.top; -(top2) -- Topic 2 -->
<!ELEMENT top3        - O  %m.top; -(top3) -- Topic 3 -->
<!ELEMENT top4        - O  %m.top; -(top4) -- Topic 4 -->
<!ELEMENT  th         - O  %m.ph; -- Topic heading -->

              <!-- Elements in Sections or Paragraphs -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT address     - -  (aline+)>
<!ELEMENT  aline      O O  %m.ph; -- Address line -->
<!ELEMENT artwork     - O  EMPTY>
<!ELEMENT dl          - -  ((dthd+, ddhd)?, (dt+, dd)*)>
<!ELEMENT  dt         - O  %m.ph; -- Definition term -->
<!ELEMENT  (dthd|ddhd)- O  (#PCDATA) -- Headings for dt and dd -->
<!ELEMENT  dd         - O  %m.pseq; -- Definition description -->
<!ELEMENT gl          - -  ((gt, (gd|gdg))*) -- Glossary list -->
<!ELEMENT  gt         - O  (#PCDATA) -- Glossary term -->
<!ELEMENT  gdg        - O  (gd+) -- Glossary definition group -->
<!ELEMENT  gd         - O  %m.pseq; -- Glossary definition -->
<!ELEMENT (%ps.ul.d;) - -  (li*) -- Unit item lists -->
<!ELEMENT  li         - O  %m.pseq; -- List item -->
<!ELEMENT lines       O O  %m.pseq; -- Line elements -->
<!ELEMENT (lq|xmp)    - -  %m.pseq; -(%i.float;) -- Long quotation -->
<!ELEMENT (%s.p.d;)   O O  %m.p; -- Paragraphs -->

                            <!-- Table -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT tbl         - -  (hr*, fr*, r+)>
<!ELEMENT  hr         - O  (h+) -- Heading row -->
<!ELEMENT  fr         - O  (f+) -- Footing row -->
<!ELEMENT  r          O O  (c+) -- Row (body of table) -->
<!ELEMENT   c         O O  %m.pseq; -- Cell in body row -->
<!ELEMENT   (f|h)     O O  (#PCDATA) -- Cell in fr or hr -->

                           <!-- Phrases -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT (%p.em.ph;) - -  %m.ph;   -- Emphasized phrases -->
<!ELEMENT q           - -  %m.ph;   -- Quotation -->
<!ELEMENT (%p.rf.ph;) - O  %m.ph;   -- Reference phrases -->
<!ELEMENT (%p.rf.d;)  - O  EMPTY    -- Generated references -->

                    <!-- Includable Subelements -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT fig         - -  (figbody, (figcap, figdesc?)?) -(%i.float;)>
<!ELEMENT  figbody    O O  %m.pseq; -- Figure body -->
<!ELEMENT  figcap     - O  %m.ph; -- Figure caption -->
<!ELEMENT  figdesc    - O  %m.pseq; -- Figure description -->
<!ELEMENT fn          - -  %m.pseq; -(%i.float;) -- Footnote -->
<!ELEMENT ix          - O  (#PCDATA) -- Index entry -->

                  <!-- Attribute Definition Lists -->
<!-- As this document type definition is intended for basic SGML
     documents, in which the LINK features are not supported, it was
     necessary to include link attributes in the definitions.
-->
```

```
<!--      ELEMENTS    NAME     VALUE     DEFAULT -->
<!ATTLIST %doctype;   security CDATA     #IMPLIED
                      status   CDATA     ""
                      version  CDATA     #IMPLIED>
<!ATTLIST title       stitle   CDATA     #IMPLIED>
<!ATTLIST (h0|h1|h2|%bm.d;|%fm.d;)
                      id       ID        #IMPLIED
                      stitle   CDATA     #IMPLIED>
<!ATTLIST (h3|h4)     id       ID        #IMPLIED>
<!ATTLIST artwork     sizex    NMTOKEN   textsize
          -- Default is current text width in column. --
                      sizey    NUTOKEN   #REQUIRED
          -- (Sizes are specified in the units supported by the
             application in which this declaration appears;
             for sizex, the keyword "textsize" can be used
             to mean "the width at which previous text was set").
          -->
<!ATTLIST gl          compact  (compact) #IMPLIED
                      termhi   NUMBER    2>
<!ATTLIST dl          compact  (compact) #IMPLIED
                      headhi   NUMBER    2
                      termhi   NUMBER    2
                      tsize    NUMBERS   9
          -- The number of dt elements per dd must equal the
             number of numbers specified for tsize (here 1).
             The number of dthd elements must be the same.
          -->
<!ATTLIST gd          source   CDATA     #IMPLIED>
<!ATTLIST (%ps.ul.d;) compact  (compact) #IMPLIED>
<!ATTLIST li          id       ID        #IMPLIED>
<!ATTLIST xmp         depth    NUTOKEN   #IMPLIED
                      keep     NMTOKEN   all
                      lines    (flow|lines) lines>
<!ATTLIST tbl         cols     NUMBERS   #REQUIRED
          -- The number of c elements per r must equal
             the number of numbers specified for cols
             (similarly, the number of h per hr and f per fr).
          -->
<!ATTLIST c           heading  (h)       #IMPLIED
          -- If h is specified, cell is row heading.
          -->
<!ATTLIST (%p.rf.ph;) refid    IDREF     #CONREF
                      page     (yes|no)  yes>
<!ATTLIST fnref       refid    IDREF     #REQUIRED>
<!ATTLIST liref       refid    IDREF     #REQUIRED
                      page     (yes|no)  yes>
<!ATTLIST fig         id       ID        #IMPLIED
                      frame    (box|rule|none)    none
                      place    (top|fixed|bottom) top
                      width    (column|page)      page
                      align    (left|center|right) center
                      lines    (flow|lines)       lines>
<!ATTLIST ix          id       ID        #IMPLIED
                      print    CDATA     #IMPLIED
                      see      CDATA     #IMPLIED
                      seeid    IDREF     #IMPLIED>
<!ATTLIST fn          id       ID        #IMPLIED>

                <!-- Entities for Short References -->
<!ENTITY   ptag   STARTTAG "p"      -- Paragraph start-tag -->
<!ENTITY   qtag   STARTTAG "q"      -- Quoted phrase start-tag -->
<!ENTITY   qetag  ENDTAG   "q"      -- Quoted phrase end-tag -->
```

```
<!ENTITY    endtag ENDTAG   ""       -- Empty end-tag for any element -->

<!SHORTREF docmap                       -- Map for general use --
                  "&#RS;&#RE;" ptag     -- Blank line is <p> --
                  '"'          qtag     -- " is <q> -->
<!USEMAP    docmap %doctype;>
<!SHORTREF qmap                         -- Map for quoted phrases --
                  '"'          qetag    -- " is </q> -->
<!USEMAP    qmap q>
<!SHORTREF ixmap                        -- Map for index entries --
                  "&#RE;"      endtag   -- Record end is </> -->
<!USEMAP    ixmap ix>
```

5.6 Commentary on the document type definition

5.6.1 Parameter entity declarations

Extensive use has been made of parameter entities, both in order to shorten the DTD and thus make it easier to read as well as to facilitate extensions by grouping elements with similar properties. Adding a unit item list type, for example, can be accomplished by merely modifying the ps.ul.d parameter entity definition.

5.6.2 Element and attribute declarations

The elements have been grouped according to type. Comments have been added, where appropriate, to assist the reader.

5.6.3 Short references

Three short reference maps are defined in the general document type definition.

Docmap maps blank lines onto the start-tag for a paragraph and maps ″ onto the start-tag of a quotation. This map is active for all elements except in quotations and in index entries, where element specific short reference maps are in effect. Thus

```
 ... type definition.

Docmap maps ...
```

is interpreted as

```
 ... type definition.
<p>
Docmap maps ...
```

and

```
He said "Hello ...
```

is interpreted as

```
He said <q>Hello ...
```

Qmap maps ″ onto the end-tag of a quotation. This map is active in the quotation element. Thus

```
He said "Hello! How are you?"
```

is interpreted as

```
He said <q>Hello! How are you?</q>
```

Docmap is active at the start of the line and maps the first ″ into <q>, changing the current map to qmap. This map is active when the second ″ is encountered and this character is thus mapped onto </q>.

Ixmap maps record end onto an end-tag. This map is active in the ix element and this element is thus ended, at the latest, at the end of the record containing the ix start-tag.

5.7 National versions of the elements of the general document

It is sometimes desirable that the generic identifiers reflect terms used in national languages. This is the case, for example, for documents of user communities confined to one language and where the exchange of documents with outside users is small. The document components identified above, however, have direct equivalents in many other languages and thus a one to one mapping onto a selection of other languages is given below. The English generic identifiers and attributes are shown in parentheses for easier reference to the description of the elements above.

5.7.1 French

document général	docgén (general)
	class= (security=)
	état= (status=)
	version= (version=)

partie liminaire	limin (frontm)
corps de document	corps (body)
annex	annex (appendix)
partie terminale	termin (backm)
table des matières	tdm (toc)
liste des figures	lfig (figlist)
index	index (index)
page de titre	pagtit (titlep)
titre du document	titre (title) rtitre= (stitle=)
référence du document	réfdoc (docnum)
date	date (date)
auteur	auteur (author)
adresse	adresse (address)

résumé	résumé (abstract)
	id= (id=)
	rtitre= (stitle=)

préface	préface (preface)
	id= (id=)
	rtitre= (stitle=)

glossaire	gloss (glossary)	
	id= (id=)	
	rtitre= (stitle=)	
bibliographie	bibliog (bibliog)	
	id= (id=)	
	rtitre= (stitle=)	
partie de niveau 0 - 4	p0, p1, p2 (h0, h1, h2)	
	id= (id=)	
	rtitre= (stitle=)	
	p3, p4 (h3, h4)	
	id= (id=)	
élément de type 1-4	élé1, élé2, élé3, élé4 (top1, top2, top3, top4)	
titre d'élément	télé (th)	
tableau	tbl (tbl)	
	ncols= (cols=)	
tête de tableau	tt (hr)	
tête de colonne	tc (h)	
pied de tableau	pt (fr)	
pied de colonne	pc (f)	
ligne du tableau	l (r)	
élément d'une ligne	é (c)	
	rubr= (heading=)	
figure	fig (fig)	
	id= (id=)	
	delim= (frame=)	cadre bordure absent
	position= (place=)	haut fixe bas
	largeur= (width=)	colonne page
	aligner= (align=)	gauche centré droite
	lignes= (lines=)	contin lignes
corps de figure	figcorps (figbody)	
titre de figure	figtit (figcap)	
légende de figure	figleg (figdesc)	
réserve	réserve (artwork)	
	larg= (sizex=)	
	haut= (sizey=)	
mises en valeur 0-3	mv1, mv2, mv3, mv0 (hp1, hp2, hp3, hp0)	
référance	réf (cit)	
citation	cit (q)	

citation longue	citl (lq)	
liste de définitions	ld (dl)	
	compact (compact=)	compact
	rubacc= (headhi=)	
	termacc= (termhi=)	
	terlarg= (tsize=)	
titre de la colonne des termes	tct (dthd)	
titre de la colonne de définitions	tcd (ddhd)	
terme défini	td (dt)	
définition	déf (dd)	
liste du glossaire	lg (gl)	
	compact (compact=)	compact
	termacc= (termhi=)	
terme du glossaire	tg (gt)	
ensemble des définitions d'un terme du glossaire	edg (gdg)	
définition d'un terme du glossaire	dg (gd)	
	origine= (source=)	
liste ordonnée	lo (ol)	
	compact (compact=)	compact
liste simple	ls (sl)	
	compact (compact=)	compact
liste non ordonnée	ln (ul)	
	compact (compact=)	compact
liste de notes	lnotes (nl)	
	compact (compact=)	compact
item	il (li)	
	id= (id=)	
référence à une partie	pref (hdref)	
	idref= (refid=)	
	page= (page=)	oui non
référence à une figure	reffig (figref)	
	idref= (refid=)	
	page= (page=)	oui non
appel de note de bas de page	refnb (fnref)	
	idref= (refid=)	

référence à un item	refil (liref)	
	idref= (refid=)	
	page= (page=)	oui non
paragraphe	p (p)	
note	note (note)	
exemple	exe (xmp)	
	prof= (depth=)	
	garder= (keep=)	
	lignes= (lines=)	contin lignes
note en bas de page	nb (fn)	
	id= (id=)	
termes indexés	ix (ix)	
	id= (id=)	
	indiquer= (print=)	
	voir= (see=)	
	idvoir= (seeid=)	

5.7.2 German

Allgemeines Dokument	allgdok (general)
	schutz= (security=)
	stand= (status=)
	version= (version=)
vorderer Teil	vordertl (frontm)
Hauptteil	haupttl (body)
Anhang	anhang (appendix)
hinterer Teil	hinttl (backm)
Inhaltsverzeichnis	inhverz (toc)
Abbildungsverzeivchnis	abbverz (figlist)
Stichwortverzeichnis	swverz (index)
Titelseite	titels (titlep)
Titel (des Dokuments)	titel (title)
	kurz= (stitle=)
Dokumentnummer	doknum (docnum)
Datum	datum (date)
Autor	autor (author)
Anschrift	anschrft (address)
Kurzfassung	kurzfsg (abstract)
	k= (id=)
	kurz= (stitle=)

Vorwort	vorwort (preface)	
	k= (id=)	
	kurz= (stitle=)	
Glossar	glossar (glossary)	
	k= (id=)	
	kurz= (stitle=)	
Litteraturverzeichnis	litverz (bibliog)	
	k= (id=)	
	kurz= (stitle=)	
Überschrift Stufe 0-4	ue0, ue1, ue2 (h0, h1, h2)	
	k= (id=)	
	kurz= (stitle=)	
	ue3, ue4 (h3, h4)	
	k= (id=)	
Thema der Stufe 1-4	thema1, thema2, thema3, thema4 (top1, top2, top3, top4)	
Themenüberschrift	themue (th)	
Tabelle	tbl (tbl)	
	spalten= (cols=)	
Kopfzeile	kz (hr)	
Feld in Kopfzeile	kzf (h)	
Fußzeile	fz (fr)	
Feld in Fußzeile	fzf (f)	
Zeile in Haupttabellentail	zl (r)	
Feld in Zeile	f (c)	
	kopf= (heading=)	
Abbildung	abb (fig)	
	k= (id=)	
	rahmen= (frame=)	rahmen linien keiner
	ort= (place=)	oben fest unten
	breite= (width=)	spalte seite
	ausricht= (align=)	links mitte rechts
	zeilen= (lines=)	fliess zeilen
Abbildungshauptteil	abbh (figbody)	
Abbildungsunterschrift	abbu (figcap)	
Abbildungsbeschreibung	abbb (figdesc)	
einsetzbares Bild	bild (artwork)	
	breite= (sizex=)	
	hoehe= (sizey=)	
Hervorhebung 0-3	hv1, hv2, hv3, hv0 (hp1, hp2, hp3, hp0)	

Litteraturverweis	litvw (cit)	
kurzes Zitat	z (q)	
langes Zitate	lz (lq)	
Definitionsliste	dl (dl)	
	eng= (compact=)	kompakt
	kopfhv= (headhi=)	
	worthv= (termhi=)	
	wlaenge= (tsize=)	
Definitionen: Worteingtragkopf	dwk (dthd)	
Definitionen: Beschreibungskopf	dbk (ddhd)	
Definitionsworteintrag	dw (dt)	
Definitionsbeschreibung	db (dd)	
Glossarliste	gl (gl)	
	eng= (compact=)	kompakt
	worthv= (termhi=)	
Glossarworteintrag	gw (gt)	
Glossarbedeutungsgruppe	gbg (gdg)	
Glossarbedeutung	gb (gd)	
	quelle= (source=)	
geordnete Liste	ol (ol)	
	eng= (compact=)	kompakt
einfache Liste	el (sl)	
	eng= (compact=)	kompakt
ungeordnete Liste	ul (ul)	
	eng= (compact=)	kompakt
Anmerkungsliste	al (nl)	
	eng= (compact=)	kompakt
Listenelement	le (li)	
	k= (id=)	
Verweis auf Überschrift	uevw (hdref)	
	vwk= (refid=)	
	seite= (page=)	ja nein
Abbildungsverweis	abbvw (figref)	
	vwk= (refid=)	
	seite= (page=)	ja nein

Fußnotenverweis	fnvw (fnref)	
	vwk= (refid=)	
Listenelementverweis	levw (liref)	
	vwk= (refid=)	
	seite= (page=)	ja nein
Absatz	a (p)	
Anmerkung	anm (note)	
Beispiel	bsp (xmp)	
	tiefe= (depth=)	
	halte= (keep=)	
	zeilen= (lines=)	fliess zeilen
Fußnote	fn (fn)	
	k= (id=)	
Verzeichniseintrag	vze (ix)	
	k= (id=)	
	druck= (print=)	
	siehe= (see=)	
	siehek= (seeid=)	

5.7.3 Swedish

allmänt dokument	alldok (general)
	skyddskl= (security=)
	status= (status=)
	utgåva= (version=)
inledande del	inled (frontm)
huvuddel	huvudd (body)
bilagor	bilagor (appendix)
avslutande del	avslut (backm)
innehållsförteckning	innehåll (toc)
lista över illustrationer	illlista (figlist)
register	register (index)
titelsida	titels (titlep)
dokumenttitel	titel (title)
	ktitel= (stitle=)
dokument beteckning	dokbet (docnum)
datum	datum (date)
författare	författ (author)

adress	adress (address)	
sammanfattning	sammanf (abstract)	
	id= (id=)	
	krubrik= (stitle=)	
förord	förord (preface)	
	id= (id=)	
	krubrik= (stitle=)	
ordlisterubrik	ordlr (glossary)	
	id= (id=)	
	krubrik= (stitle=)	
bibliografi	bibliog (bibliog)	
	id= (id=)	
	krubrik= (stitle=)	
rubrikrang 0-4	r0, r1, r2 (h0, h1, h2)	
	id= (id=)	
	krubrik= (stitle=)	
	r3, r4 (h3, h4)	
	id= (id=)	
temarang 1-4	tema1, tema2, tema3, tema4 (top1, top2, top3, top4)	
temarubrik	tr (th)	
tabell	tab (tbl)	
	kol= (cols=)	
ledrad	lr (hr)	
spalt i ledrad	sl (h)	
fotrad	fr (fr)	
spalt i fortrad	sf (f)	
datarad	dr (r)	
datafält	df (c)	
	rubrik= (heading=)	
illustration	ill (fig)	
	id= (id=)	
	ram= (frame=)	hel halv ingen
	plats= (place=)	ovan fast nedan
	bredd= (width=)	kolumn sida
	inpassa= (align=)	vänster centrerad höger
	rader= (lines=)	löpande fasta
illustration	illustr (figbody)	
illustrationsrubrik	illrubr (figcap)	
illustrationstext	illtext (figdesc)	

ritningar	ritning (artwork)	
	xstorl= (sizex=)	
	ystorl= (sizey=)	
betonade fraser 0-3	bf1, bf2, bf3, bf0 (hp1, hp2, hp3, hp0)	
refererade titlar	ref (cit)	
kort citat	c (q)	
långt citat	lc (lq)	
ordförklaringar	of (dl)	
	packning= (compact=)	kompakt
	rubrbet= (headhi=)	
	termbet= (termhi=)	
	termstor= (tsize=)	
termrubrik	oftr (dthd)	
definitionrubrik	ofdr (ddhd)	
term	oft (dt)	
definition	ofd (dd)	
ordlista	ordl (gl)	
	packning= (compact=)	kompakt
	termbet= (termhi=)	
term	ord (gt)	
definitonsgrupp	ordg (gdg)	
definition	ordd (gd)	
	källa= (source=)	
indexerad lista	il (ol)	
	packning= (compact=)	kompakt
osorterad lista	ol (sl)	
	packning= (compact=)	kompakt
markerad lista	ml (ul)	
	packning= (compact=)	kompakt
observanda (lista)	obsl (nl)	
	packning= (compact=)	kompakt
listavsnitt	la (li)	
	id= (id=)	
rubrikreferens	rubref (hdref)	
	refid= (refid=)	
	sida= (page=)	ja nej
illustrationsreferens	illref (figref)	
	refid= (refid=)	
	sida= (page=)	ja nej

fotnotreferens	fotref (fnref) refid= (refid=)	

listavsnittsreferens	laref (liref)
	refid= (refid=)
	sida= (page=)
		ja nej

stycke	s (p)	
observandum	obs (note)	

exempel	exe (xmp)
	höjd= (depth=)
	minst= (keep=)
	rader= (lines=)
		löpande fasta

fotnot	fn (fn) id= (id=)	

registerord	ro (ix)
	id= (id=)
	skriv= (print=)
	se= (see=)
	seid= (seeid=)	

5.7.4 Danish

generelt dokument	gendok (general)
	sikkerh= (security=)
	status= (status=)
	version= (version=)

indledende del	indldel (frontm)
hoveddel	hoveddel (body)
appendiks	append (appendix)
afslutende del	slutdel (backm)
indholdsfortegnelse	indhldsf (toc)
liste over illustrationer	illliste (figlist)
register	register (index)
titelblad	titelbl (titlep)
dokument titel	titel (title) ktitel= (stitle=)
dokument nummer	doknr (docnum)
dato	dato (date)
forfatter	forf (author)

adresse	adr (address)	
resume	resume (abstract)	
	id= (id=)	
	utitel= (stitle=)	
forord	forord (preface)	
	id= (id=)	
	utitel= (stitle=)	
ordforklaring	ordforkl (glossary)	
	id= (id=)	
	utitel= (stitle=)	
bibliografi	bibliog (bibliog)	
	id= (id=)	
	utitel= (stitle=)	
overskrift niveau 0-4	o0, o1, o2 (h0, h1, h2)	
	id= (id=)	
	utitel= (stitle=)	
	o3, o4 (h3, h4)	
	id= (id=)	
tema niveau 1-4	tema1, tema2, tema3, tema4 (top1, top2, top3, top4)	
temaoverskrift	temao (th)	
tabel	tabel (tbl)	
	kol= (cols)	
kolonnebetegnelsesrække i tabelhoved	thr (hr)	
kolonneoverskrift i kolonnebetegnelsesrække i tabelhoved	thro (h)	
kolonnebetegnelsesrække i tabelfod	tfr (fr)	
kolonneoverskrift i kolonnebetegnelsesrække i tabelfod	tfro (f)	
rækkebetegnelse i hovedtabeldel	rt (r)	
felt i række	f (c)	
	betegn= (heading=)	
illustration	ill (fig)	
	id= (id=)	
	ramme= (frame=)	kasse linie ingen
	sted= (place=)	øverst fast nederst
	bredde= (width=)	spalte side
	plac= (align)	venstre midt højre
	linier= (lines=)	flydende linie

illustrationskrop	illkrop (figbody)	
illustrationsundertitel	illutit (figcap)	
illustrationsbeskrivelse	illbeskr (figdesc)	
billede	billede (artwork)	
	strx= (sizex=)	
	stry= (sizey=)	
fremhævelse 0-3	f1, f2, f3, f0 (hp1, hp2, hp3, hp0)	
citat	cit (cit)	
kort citat	kc (q)	
langt citat	lc (lq)	
definitionsliste	dl (dl)	
	kompakt= (compact=)	kompakt
	ofh (headhi=)	
	termfh (termhi=)	
	termstr (tsize=)	
definitionstermoverskrift	dtermo (dthd)	
definitionsdefinitionoverskrift		
	ddo (ddhd)	
definitionsterm	dterm (dt)	
definitionsdefinition	dd (dd)	
ordliste	ordliste (gl)	
	kompakt= (compact=)	kompakt
	termfh= (termhi=)	
ord	ord (gt)	
orddefinitionsgruppe	orddgr (gdg)	
orddefinition	ordd (gd)	
	kilde= (source=)	
ordnet liste	ordnl (ol)	
	kompakt= (compact=)	kompakt
enkel liste	el (sl)	
	kompakt= (compact=)	kompakt
uordnet liste	uordnl (ul)	
	kompakt= (compact=)	kompakt
liste med bemærkninger	beml (nl)	
	kompakt= (compact=)	kompakt
listeelement	le (li)	
	id= (id=)	

titelhenvisning	titelh (hdref)	
	hid= (refid=)	
	side= (page=)	ja nej
illustrationshenvisning	illh (figref)	
	hid= (refid=)	
	side= (page=)	ja nej
fodnotehenvisning	fnh (fnref)	
	hid= (refid=)	
listeelementshenvisning	leh (liref)	
	hid= (refid=)	
	side= (page=)	ja nej
afsnit	a (p)	
note	note (note)	
eksempel	eks (xmp)	
	dybde= (depth)	
	behold= (keep=)	
	linier= (lines=)	flydende linie
fodnote	fn (fn)	
	id= (id=)	
registerindgang	ri (ix)	
	id= (id=)	
	skriv= (print=)	
	se= (see=)	
	seid= (seeid=)	

5.7.5 Dutch

algemeen dokument	algdoc (general)
	veilig= (security=)
	status= (status=)
	versie= (version=)
voorafgaand materiaal	voorm (frontm)
hoofd materiaal	inhoud (body)
appendix sectie	appendix (appendix)
achterafkomend materiaal	
	achterm (backm)
inhouds opgave	iho (toc)
lijst van figuren	figlijst (figlist)
index	index (index)
titel pagina	titelp (titlep)

titel (van het dokument)	titel (title)	
	ktitel= (stitle=)	
nummer van het dokument	doknum (docnum)	
datum	datum (date)	
auteur	auteur (author)	
adres	adres (address)	
samenvatting	samenvat (abstract)	
	id= (id=)	
	ktitel= (stitle=)	
voorwoord	voorw (preface)	
	id= (id=)	
	ktitel= (stitle=)	
collectie van termen	collterm (glossary)	
	id= (id=)	
	ktitel= (stitle=)	
literatuur opgave	literat (bibliog)	
	id= (id=)	
	ktitel= (stitle=)	
onderdeel 0-4	o0, o1, o2 (h0, h1, h2)	
	id= (id=)	
	ktitel= (stitle=)	
	o3, o4 (h3, h4)	
	id= (id=)	
onderwerpen 1-4	ond1, ond2, ond3, ond4 (top1, top2, top3, top4)	
onderwerp hoofd	oh (th)	
tabel	tabel (tbl)	
	kolonn= (cols=)	
hoofdrij	hr (hr)	
hoofd in hoofdrij	h (h)	
voet rij	vr (fr)	
hoofd in voet rij	v (f)	
rij in de tabel	r (r)	
cel	c (c)	
	hoofd= (heading=)	
figuur	fig (fig)	
	id= (id=)	
	kader= (frame=)	kader lijnen geen
	plaats= (place=)	boven inplaats beneden
	breedte= (width=)	kolom pagina
	richting= (align=)	links midden rechts
	lijn= (lines=)	vloeiend lijnen

inhoud van de figuur	figinh (figbody)	
titel van de figuur	figtit (figcap)	
beschrijving van de figuur	figbes (figdesc)	

bijvoegbaar kunstwerk	kunst (artwork)
	groottex= (sizex=)
	groottey= (sizey=)	

geaccentueerde frasen 0-3	gf1, gf2, gf3, gf0 (hp1, hp2, hp3, hp0)	
aanhalingen	aanh (cit)	
korte aanhaling	a (q)	
lange aanhaling	la (lq)	

lijst van definities	lvd (dl)
	kompakt= (compact=)
	naamacc= (headhi=)
	termacc= (termhi=)
	tgrootte= (tsize=)
		kompakt

naam	ndt (dthd)	
beschrijving van de definitie	ndd (ddhd)	
term	dt (dt)	
definitie van de term	dd (dd)	

lijst van termen	lvt (gl)
	kompakt= (compact=)
	termacc= (termhi=)
		kompakt

termnaam	tn (gt)	
groep van definities	gtd (gdg)	
term definitie	td (gd) bron= (source=)	
geordende lijst	gl (ol) kompakt= (compact=)	kompakt
eenvondige lijst	el (sl) kompakt= (compact=)	kompakt
ongeordende lijst	ol (ul) kompakt= (compact=)	kompakt
lijst van notities	lvn (nl) kompakt= (compact=)	kompakt

onderdeel van de lijst	odl (li)	
	id= (id=)	
referentie naar onderdeel	refno (hdref)	
	refid= (refid=)	
	pagina= (page=)	ja nee
referentie naar figuur	refnf (figref)	
	refid= (refid=)	
	pagina= (page=)	ja nee
referentie naar voetnoot	refnv (fnref)	
	refid= (refid=)	
referentie naar lijst onderdeel		
	reflo (liref)	
	refid= (refid=)	
	pagina= (page=)	ja nee
paragraaf	p (p)	
notitie	notitie (note)	
voorbeeld	vb (xmp)	
	diepte= (depth=)	
	bewaar= (keep=)	
	lijnen= (lines=)	vloeiend lijnen
voetnoot	vn (fn)	
	id= (id=)	
woord in index	ix (ix)	
	id= (id=)	
	afdruk= (print=)	
	zie= (see=)	
	zieid= (seeid=)	

5.7.6 Generic identifiers

English	French	German	Swedish	Danish	Dutch
general	docgén	allgdok	alldok	gendok	algdoc
frontm	limin	vordertl	inled	indldel	voorm
body	corps	haupttl	huvudd	hoveddel	inhoud
appendix	annex	anhang	bilagor	append	appendix
backm	termin	hinttl	avslut	slutdel	achterm
toc	tdm	inhverz	innehåll	indhldsf	iho
figlist	lfig	abbverz	illlista	illiste	figlijst
index	index	swverz	register	register	index
titlep	pagtit	titels	titels	titelbl	titelp
title	titre	titel	titel	titel	titel
docnum	réfdoc	doknum	dokbet	doknr	doknum
date	date	datum	datum	dato	datum
author	auteur	autor	författ	forf	auteur
address	adresse	anschrft	adress	adr	adres
abstract	résumé	kurzfsg	sammanf	resume	samenvat
preface	préface	vorwort	förord	forord	voorw
glossary	gloss	glossar	ordlr	ordforkl	collterm
bibliog	bibliog	litverz	bibliog	bibliog	literat
h0—h4	p0—p4	ue0—ue4	r0—r4	o0—o4	o0—o4
top1—top4	élé1—élé4	thema1—thema4	tema1—tema4	tema1—tema4	ond1—ond4
th	télé	themue	tr	temao	oh

tbl	tbl	tbl	tab	tabel	tabel
hr	tt	kz	lr	thr	hr
h	tc	kzf	sl	thro	h
fr	pt	fz	fr	tfr	vr
f	pc	fzf	sf	tfro	v
r	l	zl	dr	rt	r
c	é	f	df	f	c
fig	fig	abb	ill	ill	fig
figbody	figcorps	abbh	illustr	illkrop	figinh
figcap	figtit	abbu	illrubr	illutit	figtit
figdesc	figleg	abbb	illtext	illbeskr	figbes
artwork	réserve	bild	ritning	billede	kunst
hp0—hp3	mv0—mv3	hv0—hv3	bf0—bf3	f0—f4	gf0—gf4
cit	réf	litvw	ref	cit	aanh
q	cit	z	c	kc	a
lq	citl	lz	lc	lc	la
dl	ld	dl	of	dl	lvd
dthd	tct	dwk	oftr	dtermo	ndt
ddhd	tcd	dbk	ofdr	ddo	ndd
dt	td	dw	oft	dterm	dt
dd	déf	db	ofd	dd	dd
gl	lg	gl	ordl	ordliste	lvt
gt	tg	gw	ord	ord	tn
gdg	edg	gbg	ordg	orddgr	gtd
gd	dg	gb	ordd	ordd	td
ol	lo	ol	il	ordnl	gl
sl	ls	el	ol	ol	el
ul	ln	ul	ml	uordnl	ol
nl	lnotes	al	obsl	beml	lvn
li	il	le	la	le	odl
hdref	pref	uevw	rubref	titelh	refno
figref	reffig	abbvw	illref	illh	refnf
fnref	refnb	fnvw	fotref	fnh	refnv
liref	refil	levw	laref	leh	reflo
p	p	a	s	a	p
note	note	anm	obs	note	notitie
xmp	exe	bsp	exe	eks	vb
fn	nb	fn	fn	fn	vn
ix	ix	vze	ro	ri	ix

5.8 Extensions to the general document

In many cases a minor extension to an existing DTD is required to add or modify elements. This can, of course, be achieved by the modification of the DTD, but it can also be accomplished by the merging of new definitions and a reference to the base DTD. Two methods for the merging are shown.

5.8.1 Extending the bibliography to include a bibliography list

In this example the bibliography element is extended to include a bibliography list, consisting of bibliographic items. These in turn contain author information and cited title information.

The extension has been made by defining the following DTD, stored in the system entity EXTENDED.DTD. It merges in the general DTD by the use of an entity reference.

```
<!ENTITY % ISOgen PUBLIC "ISO 8879-1986//DTD General Document//EN" >
<!ENTITY % ps.ul.d "ol|sl|ul|nl"                                   >
<!ENTITY % ps.list "%ps.ul.d;|dl|gl|bl"  -- bl list type added -->
<!ENTITY % p.rf.d  "fnref|liref|bibref"  -- bibref added       -->
<!ENTITY % doctype "extended" >
   %ISOgen;
<!--       ELEMENTS     MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT bl            - -  (bib*)           -- bibliography list -->
<!ELEMENT bib           - O  ((#PCDATA|cit)*) -- bibl. entry       -->
<!--       ELEMENTS     NAME      VALUE      DEFAULT -->
<!ATTLIST bib           id        ID         #IMPLIED  >
<!ATTLIST bibref        refid     IDREF      #REQUIRED >
```

A typical invocation and a small bibliographic list would be

```
<!DOCTYPE extended SYSTEM "EXTENDED.DTD">
<extended>
 ...
 ...   <bibref refid=sgml>
 ...
<backm>
<h1>Bibliography
<bl>
<bib id=sgml>ISO 8879, <cit>Information processing -
Text and office systems - Standard Generalized Markup Language (SGML)
</cit>. Geneva 1986.
 ...
</bl>
</extended>
```

5.8.2 Extending the DTD to include tables

The example of the general document contains figures as a floating element. In many applications there is a requirement for tables as well as figures. The content of the tables would normally be tabular material <tbl> element), but can also be various types of lists, notes, and paragraphs.

NOTE — The <tbl> element used in this example is that of clause E.1 of ISO 8879. An element definition for more complex tables is discussed in clause 9.

The extension of the general DTD has been made in the document itself.

```
<!DOCTYPE general PUBLIC "ISO 8879-1986//DTD General Document//EN" [
  <!ENTITY % i.float "tab|fig|fn"          -- tab    added -->
  <!ENTITY % p.rf.ph "tabref|hdref|figref" -- tabref added -->
  <!ENTITY % p.em.ph "hp1|hp2|hp3|hp0|cit" >
  <!ENTITY % p.rf.d  "fnref|liref" >
  <!ENTITY % p.zz.ph "q|(%p.em.ph;)|(%p.rf.ph;)|(%p.rf.d;)" >
  <!ENTITY % ps.ul.d "ol|sl|ul|nl" >
  <!ENTITY % ps.list "%ps.ul.d;|dl|gl" >
  <!ENTITY % ps.elem "xmp|lq|lines|tbl|address|artwork" >
  <!ENTITY % ps.zz   "(%ps.elem;)|(%ps.list;)" >
  <!ENTITY % s.p.d   "p|note" >
  <!ENTITY % m.ph    "(#PCDATA|(%p.zz.ph;))*" >
  <!ENTITY % m.pseq  "(p, ((%s.p.d;)|(%ps.zz;))*)" >
  <!--       ELEMENTS     MIN  CONTENT                     (EXCEPTIONS) -->
  <!ELEMENT tab           - -  ((tabcap,tabdesc?)?,tabbody) -(%i.float;) >
  <!ELEMENT  tabcap       - O  %m.ph; -- Table caption -->
  <!ELEMENT  tabdesc      - O  %m.pseq; -- Table description -->
  <!ELEMENT  tabbody      O O  %m.pseq; -- Table body -->
  <!--       ELEMENTS     NAME      VALUE      DEFAULT -->
  <!ATTLIST tab           id        ID         #IMPLIED
                          frame     (box|rule|none)     none
```

```
                        place     (top|fixed|bottom)  top
                        width     (column|page)       page
                        align     (left|center|right) center
                        lines     (flow|lines)        lines>
]>
<general>
 ...
<tab id=sun>
<tabcap>Hours of sunshine per day during the ISO/TC 97/SC 18/WG 8
Hamburg meeting.
<tabdesc>The meeting covered the period April 26 to 30, 1987.
<tabbody>
<tbl cols=2>
<hr>Day       <h>Hours of sunshine
<r>Sunday    <c>  11.2
<r>Monday    <c>  10.3
<r>Tuesday   <c>  11.4
<r>Wednesday<c>  10.9
<r>Thursday <c>   8.5
</tbl>
</tab>
</general>
```

6 Letter and Memorandum

This clause contains an example of an SGML document type definition that may be used as the basis for either a business letter or for an inter-office memorandum. The DTD developed in this clause is not meant to be the definitive structure for either document type. Rather it provides illustrative techniques of the interaction of marked sections and parameter entity references to provide multiple document type definitions.

The minimization of some elements has been considered, but is not a prime objective of this part of the Technical Report.

6.1 Constituent parts of the documents

6.1.1 Common document elements

Name <name> salute=

This element is used to identify the sender and receiver(s) of the document. The salute attribute can be used to provide a "friendly" opening salutation. In its absence the application can use appropriate subelements of name to create a salutation.

NOTE — Since the punctuation after a salutation varies from language to language, it is provided as data in this example rather than being automatically added by the application.

Title <title>

The content of this element is the title of a person (e.g. Dr., Professor, Rev., Mr., Mrs., Miss., Ms.).

Person's family name <family>

The tag <family> should be placed before the person's family name, i.e. in English his/her surname. The ordering of the elements family and given is "and" rather than "seq" to accommodate those countries where the ordering is reversed from English.

Person's given name <given>

The tag <given> should be placed before the person's given name. In a European country this would normally be his/her first name, but might also be his/her (preferred) nickname. See also family.

Qualifications <qual>

This element may be used to indicate the holding of awarded degrees, membership of professional societies and bodies, etc.

Person's role in a company <role>

This element may be used to indicate the person's role within their company.

Address <address>

The tag <address> should be placed after all appropriate parts of the name element have been given.

Company <comp>

This element is used to indicate the name of the company employing the person.

City <city>

This element is used to indicate the name of the city or town where the person's place of employment or residence is.

Postal code <code>

This element is used to indicate the appropriate postal code or zip code for the address.

Country <country>

The <country> tag should be placed before the country name part of the address.

Address line <aline>

The <aline> tag should be placed before each other type of line in the address.

Recipients of copies <cc>

This element should contain a list of <name> elements for all publicly acknowledged recipients of the document.

Blind copies <blind>

This element should contain a list of <name> elements for recipients of the document whose possession of the document is not be acknowledged.

Body of document <body>

The tag <body> should be placed at the start of the document's text and after the "fromto" and "others" elements.

Emphasized phrases <hp1>, <hp2>, <hp3>, <hp0>
</hp1>, </hp2>, </hp3>, </hp0>

An <hp1>, <hp2>, or <hp3> tag should be placed before text that is to be emphasized. There are four different kinds of emphasis. The four kinds could be: 1 italics; 2 bold; 3 bold italics; 0 no emphasis. At the end of the word or phrase that is to be emphazised the element should be terminated by the appropriate end-tag.

List <list> </list>

This element introduces a list of items. Each item should be denoted by a list item element.

List item <li>

The tag <li> should be used to denote each item in a list.

Paragraph <p>

Each paragraph of the document body text should be preceded by a <p> tag.

Quotation <q> </q>

The tag <q> should be placed before a quotation. The quotation is terminated by the end-tag </q>.

Subject of letter or memorandum
<subject>

The content of this element should a single sentence describing the purpose of the document.

6.1.2 Elements specific to a business letter

Letter element <letter>

This tag <letter> is the first tag to be used in a letter.

Sender of document <addrsr>

This tag <addrsr> should be placed before the details of the document's sender.

Recipient of document <addrse>

This tag <addrse> should be placed before the details of the document's primary recipient. If the document is being sent to other people, they should be included in the "cc" or "blind" elements.

Closing polite phrase <adieu>

This element contains the closing phrase to be included after the body of the letter.

NOTE — Since the puctuation after a closing phrase varies from language to language, it is provided as data in this example rather than being automatically added by the application.

Post scriptum <ps>

This element contains any additional paragraphs of the letter. These paragraphs are usually considered as afterthoughts.

6.1.3 Elements specific to an inter-office memorandum

Memorandum element <memo>

This tag <memo> is the first tag to be used in a memorandum.

Primary receivers of memorandum
<receiver>

This tag <receiver> should be placed before the details of the document's author.

Originator of memorandum
<sender>

This tag <sender> should be placed before the details of the document's primary recipient. If there are additional receivers, they should be included in the "cc" or "blind" elements.

6.1.4 Alphabetical list of the tags

<address>	
<addrsr>	(letter only)
<addrse>	(letter only)
<adieu>	(letter only)
<aline>	
<blind>	
<body>	
<cc>	
<country>	
<city>	
<code>	
<comp>	
<family>	
<given>	
<hp1> </hp1>	
<hp2> </hp2>	
<hp3> </hp3>	
<hp0> </hp0>	
<letter>	(letter only)
<li>	
<list> </list>	
<memo>	(memorandum only)
<name>	salute=
<p>	
<ps>	(letter only)
<q> </q>	
<qual>	
<receiver>	(memorandum only)
<role>	
<sender>	(memorandum only)
<subject>	
<title>	

6.2 Entity set

The entities "letter" and "memo" control which of the two types of documents is being defined. If the document is a letter, then letter and memo should be defined as

```
<!ENTITY % letter "INCLUDE" >
<!ENTITY % memo "IGNORE" >
```

If, however, the document is a memorandum, then they must be declared as

```
<!ENTITY % letter "IGNORE" >
<!ENTITY % memo "INCLUDE" >
```

It is expected that when this document type definition is used in an application, the setting of these entities to IGNORE or INCLUDE will be performed transparently. The end-user of the application may be required to indicate whether a letter or memorandum is being created, and then the entities will be defined automatically.

A number of additional entities are defined:

```
<![ %letter [
<!-- Define the entities for a letter -->
<!ENTITY % fromto "addrsr|addrse">
<!ENTITY % txrxs  "(name, address)">
]]>
<![ %memo [
<!-- Define the entities for a memo -->
<!ENTITY % fromto "sender|receiver">
<!ENTITY % txrxs  "name" -- Note no () or , address now -->
]]>
```

6.3 SGML features

The discussion assumes that some features of an SGML document are available in the application process. The minimum markup minimization features clause from the SGML declaration that is appropriate is

```
            MINIMIZE
                 DATATAG    NO
                 OMITTAG    YES
                 RANK       NO
                 SHORTTAG   YES
```

6.4 Typical document type declarations

6.4.1 The document type declaration for letters

```
<!DOCTYPE letter
     SYSTEM "SGML$ENTITY_TEXTS:BREVORMEMO.DTD"
     [<!ENTITY % letter "INCLUDE">
      <!ENTITY % memo "IGNORE" >]
>
```

6.4.2 The document type declaration for memoranda

```
<!DOCTYPE memo
     SYSTEM "SGML$ENTITY_TEXTS:BREVORMEMO.DTD"
     [<!ENTITY % memo "INCLUDE">
      <!ENTITY % letter "IGNORE" >]
>
```

6.5 The common document type definition

This is the text to be found in the system entity "SGML$ENTITY_TEXTS:BREVORMEMO.DTD"

```
<!-- Appropriate entity declaration in the document type declaration
          will be used to define letter and memorandum correctly. -->
<!ENTITY % emphas "hp1|hp2|hp3|hp0" -- emphasized phrases -->
<!ENTITY % others  "cc|blind">
<!ENTITY % name.se "title|given|family|qual|role"
                                -- simple elements in name  -->
<!ENTITY % addr.se "comp|city|country|code"
```

```
                              -- simple elements in address -->
<![ %letter [
<!-- Define the entities for a letter -->
<!ENTITY % fromto "addrsr|addrse">
<!ENTITY % txrxs  "name, address">
]]>
<![ %memo [
<!-- Define the entities for a memorandum -->
<!ENTITY % fromto "sender|receiver">
<!ENTITY % txrxs  "name" -- Note no address now -->
]]>
<!--       ELEMENTS        MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT (%fromto|%others) - O (%txrxs;)+ -- The plus is required here
-->
<!ELEMENT name             - O  (title?, (given? & family), qual?, role?)>
<!ELEMENT body             - O  (subject?, p+) >
<!ELEMENT (subject|adieu)  - O  (%emphas|#PCDATA)+ >
<!ELEMENT (p|ps)           - O  (%emphas|list|q|#PCDATA)+ >
<!ELEMENT (%emphas)        - -  (#PCDATA) >
<!ELEMENT (aline|li)       - O  (#PCDATA) >
<!ELEMENT address          - O  (comp?, aline+, city, country?, code?) >
<!ELEMENT list             - -  (li+) >
<!ELEMENT q                - -  (#PCDATA) >
<!ELEMENT (%name.se)       - O  (#PCDATA) >
<!ELEMENT (%addr.se)       - O  (#PCDATA) >
<!--      ELEMENTS    NAME     VALUE     DEFAULT -->
<!ATTLIST name        salute   CDATA     #IMPLIED  >
<!--                                                             -->
<![ %letter [
<!ELEMENT  letter          - -  ((%fromto | %others)+, body, adieu, ps?)>
]]>
<![ %memo [
<!ELEMENT  memo            - -  ((%fromto | %others)+, body)>
]]>
```

6.6 A sample letter

```
<!DOCTYPE letter
     SYSTEM "SGML$ENTITY_TEXTS:BREVORMEMO.DTD"
     [<!ENTITY % letter "INCLUDE">
      <!ENTITY % memo "IGNORE" >]
>
<letter>
<addrsr>
<name>
<given>Trevor
<family>Jenkins
<qual>MTech, AMBCS
<role>Technical Manager
<address>
<comp>Paralog (UK) Ltd
<aline>1 Carthusian St
<city>LONDON
<country>United Kingdom
<code>EC1M 6EB
<addrse>
<name salute="Dear Joan,">
<title>Mrs
<given>Joan M
<family>Smith
<role>Senior Consultant
<address>
```

```
<comp>NCC
<aline>Oxford Road
<city>MANCHESTER
<country>United Kingdom
<code>M1 7ED
<body>
<subject>Letter and memorandum document type definition
<p>
This is the first attempt at the document type definitions
DTDs for letters and memoranda. I have tried to make
them as small as possible but still be useful for non-trivial texts.
They should be seen as starting points not the definitive descriptions.
<p>
I have not as yet had the opportunity or the time to finish the MOTIS
comments but I will table them at the next panel 8 meeting.
<adieu>Best wishes and regards,
</letter>
```

6.7 A sample inter-office memorandum

```
<!DOCTYPE memo
     SYSTEM "SGML$ENTITY_TEXTS:BREVORMEMO.DTD"
     [<!ENTITY % memo "INCLUDE">
      <!ENTITY % letter "IGNORE" >]
>
<memo>
<sender>
<name>
<given>Trevor
<family>Jenkins
<qual>MTech, AMBCS
<role>Technical Manager
<receiver>
<name>
<title>Mrs
<given>Joan M
<family>Smith
<role>Senior Consultant
<body>
<subject>Letter and memorandum document type definition
<p>
This is the first attempt at the document type definitions
DTDs for letters and memoranda. I have tried to make
them as small as possible but still be useful for non-trivial texts.
They should be seen as starting points not the definitive descriptions.
<p>
I have not as yet had the opportunity or the time to finish the MOTIS
comments but I will table them at the next panel 8 meeting.
</memo>
```

7 Spreadsheet

This example shows an approach to using SGML for interchange of data between two different computer spreadsheet applications. All the markup is expected to be added by the application programs; not by the users.

7.1 Constituent parts of a spreadsheet

A spreadsheet is considered to be a structure of "cell" elements, organized as a matrix of rows and columns. Each cell of this structure has a unique "cell reference" given by the intersection of a row with a column. A cell may contain either a "label" or a formula. Labels consist of character data. The elements of a formula are displayed below.

7.1.1 Elements of a spreadsheet

The recursive nature of SGML models is exploited in the definition of the spreadsheet document type definition. The elements are nested to reflect the natural way of expressing mathematical formulæ. The structure is shown in figure 10.

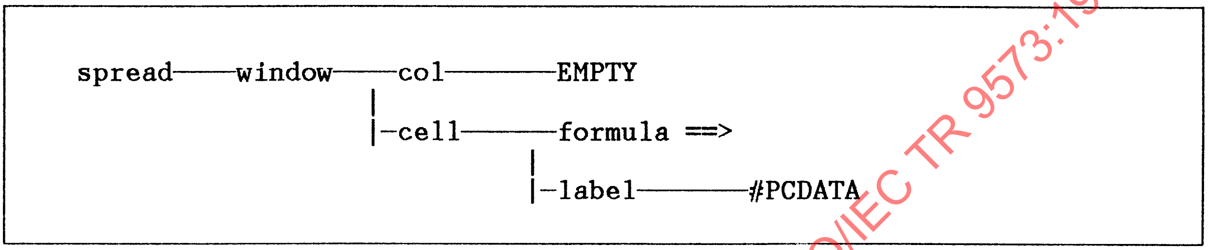


Figure 10 — Overall structure of a spreadsheet

Spreadsheet <spread> format= lformat= width= locked= name= order=

The first element that occurs in a spreadsheet is <spread>. Its attributes can be used to define settings for the format, width, and locked status of all cells in the spreadsheet. These global settings may be overridden by similiar attributes on the window, column or cell elements. The default is that all cells are not locked.

The order attribute indicates whether data entry is in row-major or column-major order. Its value affects the defaulting assumptions for the row and col identification attributes of cell elements (discussed below).

Sub-spreadsheet <window> format= lformat= width= locked= name=

A spreadsheet may be subdivided into windows. The use of this element allows settings of format, width, and locked to be overridden. The new values will remain in effect until a subsequent <window> tag is found.

Column <col> col= format= lformat= width= locked=

The tag <col> may be used to change the settings of the format, width, and lock of the column identified by the col attribute. The cell reference naming scheme used by this application identifies columns with a letter (or pair of letters) and rows with a number. The value of the col attribute is therefore a letter. (The SGML ID/IDREF mechanism is not used for spreadsheet cell identification because its requirement for uniqueness would prevent more than one spreadsheet from occurring in the same document. Also, the scheme used here allows more efficient markup minimization.)

Cell <cell> row= col= locked=

The tag <cell> identifies an individual cell of the spreadsheet. It must be followed by either a formula or a label tag. If an application using this document type definition allows variable width cells to occur in the same column, it should add the width attribute to the cell attribute list.

A cell is identified by its row, which is a number, and its column, which is a letter or pair of letters. When the document is in row-major order, the application assumes that if the row is not specified, it is the same as the row of the preceeding cell. If the column is not specified, it is assumed to be the next successive column. These assumptions would be reversed if the order attribute were column-major.

Character data <label> lformat=

This element is used when a cell contains character data rather than a formula. The appropriate label formats are left, middle, and right.

7.1.2 Elements of a formula

The method used in this example is to identify the different terms and operators of a formula with tags. Short references are employed to map commonly used keyboarding conventions into tags for the operators. Alternatively the content of a formula could be defined as CDATA with an appropriate data content notation; e.g. that of a programming language. (Note that the formula definition used here differs from that in clause 8 because the use of formulas in spreadsheets differs in many respects from the use of formulas in scientific text. There is considerable overlap, however, and ideas from either approach could be used.)

The structure of a formula is shown in figure 11.

Mathematical formula <formula> format=

The tag <formula> must be used in a cell when the cell does not contain a label.

Signed expression <sexp>

The tag <sexp> is used to start a signed expression with a formula.

Parenthesized expression <pexp>

This element is intended to allow mathematical expressions to be grouped.

Mathematical function <function> funcname=

This element specifies that the application should apply the mathematical function whose name is given as the value of the funcname attribute. It is the responsibility of the application to provide the functions that may be named.

The tag <function> can be followed by optional elements for the arguments to the function. The creator of the spreadsheet document is responsible for ensuring that the number of arguments is correct and that they are suitable for passing to the function. The application could be expected to provide some diagnostic message if an argujment is unsuitable.

Expression structure <term>
<primary>
<factor>

These elements are recursive to ensure that the precedence of the operators is in accordance with their normal mathematical usage.

Block of cells <range>

This element is used when a function is to be applied to more than one cell. The spreadsheet document's creator must ensure that a range is acceptable as an argument to the named function.

```
formula---sexp------term-----primary----factor----ref-------#PCDATA
        |         |         |         |         |
        |         |         |         |         |-#PCDATA
        |         |         |         |         |
        |         |         |         |         |-function  ==>
        |         |         |         |         |
        |         |         |         |         |-pexp------formula ..
        |         |         |         |
        |         |         |         |-expon-----EMPTY
        |         |         |
        |         |         |-%multop---multiply--EMPTY
        |         |                   |
        |         |                   |-divide----EMPTY
        |         |                   |
        |         |                   |-modulus---EMPTY
        |         |                   |
        |         |                   |-and-------EMPTY
        |         |
        |         |-%addop----add-------EMPTY
        |                   |
        |                   |-sub-------EMPTY
        |                   |
        |                   |-or--------EMPTY
        |
        |-%compop---greater---EMTPY
                  |
                  |-greateq---EMPTY
                  |
                  |-lessthan--EMPTY
                  |
                  |-lesseq----EMPTY
                  |
                  |-equal-----EMPTY
                  |
                  |-notequal--EMPTY
function---ref-------#PCDATA
         |
         |-range-----ref-------#PCDATA
         |
         |-function ...
         |
         |-constant--#PCDATA
         |
         |-pexp------formula ...
```

Figure 11 — Structure of a formula in a spreadsheet

Mathematical operators

<multiply>
<divide>
<add>
<sub>
<expon>
<modulus>

These elements represent the mathematical operations of multiplication, division, addition, subtraction, exponentiation, and remainder.

Boolean operators <and>
<or>

These elements represent the boolean operations of and and or.

Comparison operators <lessthan>
<lesseq>
<greater>
<greateq>
<equal>
<notequal>

These elements are used when the formula in a cell is to include a comparison.

Cell reference <ref>

This element is used when the content of a cell is to be used in some part of a formula. The content of the ref element is the cell identifier (row/col) of the referenced cell.

7.1.3 Alphabetical list of the tags

<add>	
<and>	
<arg>	
<args>	
<cell>	row= col= width= locked=
<col>	col= format= lformat= width= locked=
<divide>	
<equal>	
<expon>	
<factor>	
<formula>	format=
<function>	funcname=
<greateq>	
<greater>	
<label>	lformat=
<lesseq>	
<lessthan>	
<list>	
<modulus>	
<multiply>	
<notequal>	
<pexp>	
<or>	
<primary>	
<range>	
<ref>	
<sexp>	
<spread>	format= lformat= width= locked= name=
<sub>	
<term>	
<window>	format= lformat= width= locked= name=

7.1.4 Entity sets

7.1.4.1 Parameter entities

Parameter entities have been used in the document type definition to reduce the incidence of keying errors in the DTD itself and to emphasize that some constructs are used more than once. The parameter entities are:

- addop
- compop
- format
- lformat
- formsiz
- funcs
- multop

The funcs entity also allows an application to permit more function names for the funcname attribute of the function element.

7.1.4.2 Entities used with short references

The document type defintion has defined a number of entities that are specifically for use with short references. Most of these contain STARTTAG bracketed text.

7.1.4.3 Short references, notations, maps, etc.

Consideration has been given to some short references, but this has not been a prime design criterion. However, a number of short references have been defined to demonstrate how the size of the document might be reduced. Most of the short references are used via a map on the formula element. However, the cell element has a map that allows typical spreadsheet conventions to replace most label and formula tags. Also, the range element has a specialized map to allow the ":" to separate the first reference from the second, which is customary in many spreadsheets. The specialized map for range allows a result that would normally require the DATATAG feature; by using short references instead, the spreadsheet can occur in basic SGML documents.

7.1.5 SGML features

The document type defined in this clause assumes that some features are available in the SGML parser. The markup minimization features in the SGML declaration that are required are:

```
MIMIMIZE
    DATATAG  NO
    OMITTAG  YES
    RANK     NO
    SHORTTAG YES
```

NOTE — It is permissible for the application to specify YES for RANK and/or DATATAG, but they are not required for spreadsheets.

7.2 The document type definition

```
<!ENTITY % addop   "add|sub|or">
<!ENTITY % compop  "greater|greateq|lessthan|lesseq|equal|notequal">
<!ENTITY % format  "format (integer|expon|general|dollar|number) general">
<!ENTITY % lformat "lformat (left|middle|right) left">
<!ENTITY % formsiz
         "%format; %lformat; width NUMBER #IMPLIED  locked (yes|no|y|n)
no">
<!ENTITY % funcs "sin|cos|sum|percent" >
<!ENTITY % multop "multiply|divide|modulus|and">

<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT spread      - O  (window+) >
```

```
<!ELEMENT window      0 0  (col*, cell+) >
<!ATTLIST (spread|window)
                      name     NMTOKEN   #IMPLIED
                      order    (rowmajor|colmajor) rowmajor
                      %formsiz;>
<!ELEMENT col         - 0  EMPTY >
<!ATTLIST col         col      NAME      #REQUIRED -- Column letter --
                      %formsiz;>
<!ELEMENT cell        - 0  (formula|label) >
<!ATTLIST cell        row      NUMBER    #IMPLIED -- Default: same --
                      col      NAME      #IMPLIED -- Default: next --
                      locked   (yes|no|y|n) no>
<!ELEMENT label       - 0  (#PCDATA) >
<!ATTLIST label       %lformat;>

<!ELEMENT formula     0 0  (sexp, ((%compop;), sexp)?) >
<!ATTLIST formula     %format;>
<!ELEMENT (%compop)   - 0  EMPTY >
<!ELEMENT  sexp       0 0  (term, ((%addop;), term)?) >
<!ELEMENT  (%addop;)  - 0  EMPTY >
<!ELEMENT   term      0 0  (primary, ((%multop;), primary)*) >
<!ELEMENT   (%multop;)- 0  EMPTY >
<!ELEMENT    primary  0 0  (factor, (expon, factor)?) >
<!ELEMENT     expon   - 0  EMPTY >
<!ELEMENT     factor  0 0  (#PCDATA|ref|function|pexp) >
<!ELEMENT      pexp   - 0  (formula) >
<!ELEMENT      ref    0 0  (#PCDATA) -- COL/ROW of referenced cell -->

<!ELEMENT function    - -  (constant|ref|range|function|pexp)* >
<!ATTLIST function    funcname (%funcs) #REQUIRED >
<!ELEMENT  constant   - 0  (#PCDATA) >
<!ELEMENT  range      - 0  (ref, ref) >

<!-- Declarations relating to RANGE short references -->
<!ENTITY   thru     "</REF><REF>" >
<!SHORTREF range    ":" thru>
<!USEMAP   range    range>

<!-- Declarations relating to CELL short references -->
<!ENTITY llabel   STARTTAG "label" >
<!ENTITY rlabel   STARTTAG "label format=right" >
<!ENTITY formula  STARTTAG "formula" >
<!SHORTREF cellmap
      '"'   llabel
      "'"   rlabel
      "+"   formula
>
<!USEMAP  cellmap   cell>

<!-- Declarations relating to FORMULA short references -->
<!ENTITY add      STARTTAG "add" >
<!ENTITY minus    STARTTAG "sub" >
<!ENTITY divide   STARTTAG "divide" >
<!ENTITY multiply STARTTAG "multiply" >
<!ENTITY expon    STARTTAG "expon" >
<!ENTITY lparen   STARTTAG "pexp" >
<!ENTITY rparen   ENDTAG   "pexp" >
<!ENTITY equal    STARTTAG "equal" >
<!ENTITY notequal STARTTAG "notequal" >
<!ENTITY mod      STARTTAG "modulus" >
<!ENTITY or       STARTTAG "or" >
<!ENTITY sine     STARTTAG "function sin" >
```

```
<!ENTITY cosine   STARTTAG "function cos" >
<!ENTITY sum      STARTTAG "function sum" >
<!ENTITY percent  STARTTAG "function percent" >
<!SHORTREF signs
      "+"  add
      "-"  minus
      "*"  multiply
      "¬"  expon
      "%"  mod
      "="  equal
      "("  lparen
      ")"  rparen
      "#"  notequal
>
<!USEMAP  signs    formula>
```

7.3 A sample spreadsheet

This spreadsheet

Paralog coster	Embedded mode		Detached mode		Organic mode	
Lines code (K)	120					
Annual change	.12					
Person-months	1125	97	639	55	366	32
Time to develop	24	2.04	24	2.07	20	1.70
Number of people	48	4	27	2	19	2
Lines/Person month	107	1407	188	2477	328	4329
Maintenance effort	135	12	77	7	44	4
Multipliers	.09	.08				
Cost driver	VL	L	N	H	VH	EH
RELY	.75	.88	1.00	1.15	1.40	
DATA		.94	1.00	1.08	1.16	
CPLX	.70	.85	1.00	1.15	1.30	1.65
TIME			1.00	1.11	1.30	1.66
STOR			1.00	1.06	1.21	1.56
VIRT		.87	1.00	1.15	1.30	
TURN		.87	1.00	1.07	1.15	
ACAP	1.46	1.19	1.00	.86	.71	
AEXP	1.29	1.13	1.00	.91	.82	
PCAP	1.42	1.17	1.00	.86	.70	
VEXP	1.21	1.10	1.00	.90		
LEXP	1.14	1.07	1.00	.95		
MODP	1.24	1.10	1.00	.91	.82	
TOOL	1.24	1.10	1.00	.91	.83	
SCED	1.23	1.08	1.00	1.04	1.10	
MODP2K	1.25	1.12	1.00	.90	.81	
MODP8K	1.30	1.14	1.00	.88	.77	
MODP32K	1.35	1.16	1.00	.86	.74	
MODP128K	1.40	1.18	1.00	.85	.72	
MODP512K	1.45	1.20	1.00	.84	.70	

was specified using the following document:

```
<!DOCTYPE spread SYSTEM "spread.dtd">
<spread format=integer>
```

```
<cell row=1 col=A>"Paralog coster
<cell col=B>"Embedded mode
<cell col=D>"Detached mode
<cell col=F>"Organic mode
<cell row=4 col=A>"Lines code (K)
<cell>+120
<cell row=5 col=A>"Annual change
<cell format=dollar>+.12
<cell row=7 col=A>"Person-months
<cell>+3.6*(<ref/B4/¬1.2)
<cell>+<ref/B7/*<ref/B14/
<cell>+3*(<ref/B4/¬1.12)
<cell>+<ref/D7/*<ref/B14/
<cell>+2.4*(<ref/B4/¬1.05)
<cell>+<ref/F7/*<ref/B14/
<cell row=8 col=A>"Time to develop
<cell>+2.5*(<ref/B7/¬0.32)
<cell format=dollar>+<ref/B8/*<ref/B14/
<cell>+2.5*(<ref/D7/¬0.35)
<cell format=dollar>+<ref/D8/*<ref/B14/
<cell>+2.5*(<ref/F7/¬0.35)
<cell format=dollar>+<ref/F8/*<ref/B14/
<cell row=9 col=A>"Number of people
<cell>+<ref/B7/<divide><ref/B8/
<cell>+<ref/B9/*<ref/B14/
<cell>+<ref/D7/<divide><ref/D8/
<cell>+<ref/D9/*<ref/B14/
<cell>+<ref/F7/<divide><ref/F8/
<cell>+<ref/F9/*<ref/B14/
<cell row=10 col=A>"Lines/Person month
<cell>+(<ref/B4/*1000)<divide><ref/B7/
<cell>+(<ref/B4/*1000)<divide>(<ref/B7/*<ref/C14/)
<cell>+(<ref/B4/*1000)<divide><ref/D7/
<cell>+(<ref/B4/*1000)<divide>(<ref/D7/*<ref/C14/)
<cell>+(<ref/B4/*1000)<divide><ref/F7/
<cell>+(<ref/B4/*1000)<divide>(<ref/F7/*<ref/C14/)
<cell row=11 col=A>"Maintenance effort
<cell>+1*<ref/B5/*<ref/B7/
<cell>+1*<ref/B5/*<ref/C7/
<cell>+1*<ref/B5/*<ref/D7/
<cell>+1*<ref/B5/*<ref/E7/
<cell>+1*<ref/B5/*<ref/F7/
<cell>+1*<ref/B5/*<ref/G7/
<window format=dollar>
<cell row=14 col=A>"Multipliers
<cell>+<ref/B17/*<ref/F26/*<ref/E28/*
 <ref/F34/*<ref/F30/*<ref/E27/*<ref/F29/*
 <ref/E27/*<ref/F25/*<ref/F24/*<ref/B19/
<cell>+<ref/B17/*<ref/F26/*<ref/E28/*
 <ref/F34/*<ref/F30/*<ref/E27/*<ref/F36/*
 <ref/E27/*<ref/F25/*<ref/F24/*<ref/B19/
<window locked=yes format=dollar>
<cell row=16 col=A>"Cost driver
<cell>'VL
<cell>'L
<cell>'N
<cell>'H
<cell>'VH
<cell>'EH
<cell row=17 col=A>"RELY
<cell>+0.75
<cell>+0.88
```

```
<cell>+1
<cell>+1.15
<cell>+1.4
<cell row=18 col=A>"DATA
<cell col=C18>+.94
<cell col=D18>+1
<cell>+1.08
<cell>+1.16
<cell row=19 col=A>"CPLX
<cell>+0.70
<cell>+.85
<cell>+1
<cell>+1.15
<cell>+1.3
<cell>+1.65
<cell row=20 col=A>"TIME
<cell col=D20>+1
<cell>+1.11
<cell>+1.3
<cell>+1.66
<cell row=21 col=A>"STOR
<cell col=D21>+1
<cell>+1.06
<cell>+1.21
<cell>+1.56
<cell row=22 col=A>"VIRT
<cell col=C22>+.87
<cell>+1
<cell>+1.15
<cell>+1.3
<cell row=23 col=A>"TURN
<cell col=C23>+.87
<cell>+1
<cell>+1.07
<cell>+1.15
<cell row=24 col=A>"ACAP
<cell>+1.46
<cell>+1.19
<cell>+1
<cell>+.86
<cell>+.71
<cell row=25 col=A>"AEXP
<cell>+1.29
<cell>+1.13
<cell>+1
<cell>+.91
<cell>+.82
<cell row=26 col=A>"PCAP
<cell>+1.42
<cell>+1.17
<cell>+1
<cell>+.86
<cell>+.70
<cell row=27 col=A>"VEXP
<cell>+1.21
<cell>+1.10
<cell>+1
<cell>+.90
<cell row=28 col=A>"LEXP
<cell>+1.14
<cell>+1.07
<cell>+1
```

```
<cell>+.95
<cell row=29 col=A>"MODP
<cell>+1.24
<cell>+1.1
<cell>+1
<cell>+.91
<cell>+.82
<cell row=30 col=A>"TOOL
<cell>+1.24
<cell>+1.1
<cell>+1
<cell>+.91
<cell>+.83
<cell row=31 col=A>"SCED
<cell>+1.23
<cell>+1.08
<cell>+1
<cell>+1.04
<cell>+1.1
<cell row=33 col=A>"MODP2K
<cell>+1.25
<cell>+1.12
<cell>+1.
<cell>+.9
<cell>+.81
<cell row=34 col=A>"MODP8K
<cell>+1.3
<cell>+1.14
<cell>+1.00
<cell>+.88
<cell>+.77
<cell row=35 col=A>"MODP32K
<cell>+1.35
<cell>+1.16
<cell>+1
<cell>+.86
<cell>+.74
<cell row=36 col=A>"MODP128K
<cell>+1.4
<cell>+1.18
<cell>+1
<cell>+.85
<cell>+.72
<cell row=37 col=A>"MODP512K
<cell>+1.45
<cell>+1.2
<cell>+1
<cell>+.84
<cell>+.7
</spread>
```

8 Mathematics

Over the last decade, several schemes for a "language" to describe mathematical formulae have been proposed and implemented. The best known schemes include: EQN[1], TeX[2], and STI[3]. The EQN syntax has been implemented in many systems. In this application of SGML to describe mathematics, extensive use has been made of the names for "commands" used in this formalism when selecting names of elements.

8.1 Concepts

The hierarchical nature of mathematical expressions allows direct mapping on to the hierarchical nature of SGML elements. The formalism in this clause makes full use of this fact, even if, as a result, the number of different elements is sometimes increased. The structures identified are ideally suited to syntax-directed editing and display.

There are, however, certain exceptions to this rigid mapping. These exceptions include mathematical operations and operators that have a direct equivalent on most keyboards; for example =, +, and —. A user would expect to type

```
x = a + b
```

to represent "the simple sum of a and b followed by an assignment to x" rather than an expression like

```
<assign><to>x<value><simsum><term>a<term>b</simsum></assign>
```

8.2 User-interface

It is clear that all these "languages" to describe mathematics are perfectly well suited for machine to machine communication. In keyboarding mathematics on today's workstations and PCs with All Points Addressable screens, a better user interface would be appreciated.

- Immediate feedback: the formula should appear, on the screen, as it is typed in, with correct fonts, at the correct size (sometimes rescaling "on the fly").
- Assisted typing by automatic cursor positioning: for example, if a sum is being keyed, then after defining it to be a sum the cursor should be positioned at the lower limit; a "tab" would move it to the place for the upper limit, and a further "tab" would move it to the expression for the sum.
- Structure of the formula kept so modifications can be made. An application of SGML as an interchange format has obvious advantages.

This application has been designed with the aim of a one-to-one correspondence of the tags and "function keys" used in such a system, but keeping the generic identifiers as mnemonic and natural as possible, also to enable non-assisted keyboarding, which for the time being will be the case for most keyboarders.

[1] B.W. Kernighan and L.L. Cherry. *A System for Typesetting Mathematics*; CACM, Vol. 18, No. 3, March 1975. B.W. Kernighan and L.L. Cherry. *Typesetting Mathematics — User's Guide*; second edition. Bell Labs Report, 1978.

[2] Donald E. Knuth, *TAU EPSILON CHI, a system for technical text*, American Mathematical Society, Providence, 1979. Donald E. Knuth, *The TeXbook*, Addison-Wesley, 1984.

[3] *Reference Manual for Copy Editors, Production Assistants, and Keyboarders in Computer-Assisted Photocomposition*; Science Typographers, Inc.

8.3 Symbols

One of the difficulties in describing mathematics is caused by the extensive use of special symbols, normally not available on computer keyboards. This application uses the entity references in annex D of ISO 8879 to solve this problem.

8.4 Description of the elements

8.4.1 Formula and formula reference

The examples of the different elements of mathematics shown below have been simplified to show the principal uses of the tags identifying the components. The definition in 8.7 provides for extensive element nesting, and some examples are shown in 8.6.

NOTE — The indentation and line-breaks in the examples are merely to clarify the structure of the formula. It is assumed that the text formatter ignores all spaces and that all positioning is performed according to the conventions of mathematical typesetting.

Inline formula <f> </f>

This tag is used to identify an inline formula. Both the start and end-tags are required.

```
Text <f><frac>a<over>b</frac></f> more text.
```

resulting in

Text $\frac{a}{b}$ more text.

Display formula <df> </df> id= align= num=

This tag is used to identify a display formula. Both the start and end-tags are required. The value of the identifier attribute (id=) can be used when referring to a formula and must be a unique reference in the document. The alignment attribute (align=) takes one of the values: left, right, centre. Left is the default value. The number attribute (num=) can be used to specify an explicit formula number; if omitted, sequential numbering of the formulae would normally be performed by the text formatter.

```
Text <df><frac>a<over>b</frac></df> more text.
```

resulting in

Text

$$\frac{a}{b} \qquad (1)$$

more text.

Display formula group <dfg> </dfg> id= align= num=

This tag is used to identify a display formula group. Both the start and end-tags are required. The value of the identifier attribute (id=) can be used when referring to a formula group and must be a unique reference in the document. The alignment attribute (align=) takes one of the values: left, right, centre. Left is the default value. The number attribute (num=) can be used to specify an explicit formula number for the group; if omitted, sequential numbering of the formulae and formula groups would normally be performed by the text formatter.

The content of a formula group is display formulae.

NOTE — Numbering of formulae in a group would typically be a letter appended to the (typically numeric) number of the formula group. The num= attribute on a formula in the group applies to the appended number, whereas the num=

attribute on the display formula group applies only to the number of the group.

Formula reference <dfref> page= refid=

The formula reference is used within the document to refer to the identified formula or formula group. The appropriate value for the reference identifier attribute (refid=) must be provided. The page number attribute can take the values yes and no; in the first case, which is the default, the appropriate page number, supplied by the system, is added to the reference. The tag is self-contained, i.e. it has no content.

```
<df id=x><frac>a<over>b</frac></df>
<dfref refid=x page=no> is a simple example.
```

resulting in

$$\frac{a}{b} \qquad (2)$$

```
(2) is a simple example.
```

8.4.2 Formula content

Mark and markref — for horizontal alignment
<mark> id=
<markref> refid=

In many cases, for example when aligning multiple formulae on equal signs, marking and referring to a horizontal position is required. The mark tag is used to define a mark. The required identifier attribute (id=) is used when referring to a mark and must be a unique reference in the document. The tag is self-contained, i.e. it has no content.

The mark reference tag is used to reference a defined mark and causes the system to align the formula horizontally on that mark. The appropriate value for the reference identifier attribute (refid=) must be provided. The tag is self-contained, i.e. it has no content.

```
<df> z <mark id=x> = x + 6y - 3 + 12x - 3y</df>
<df> <markref refid=x> = 8y - 12 </df>
<df> <markref refid=x> = y - 3 </df>
```

resulting in

$$\begin{aligned} z &= x + 6y - 3 + 12x - 3y \\ &= x + 3y - 3 + 12x \\ &= 13x + 3y - 3 \end{aligned}$$

Break — for division points in formula
<break> type=

The break point tag is used to define breakpoints in a formula. The type attribute (type=) with values optional or required indicates if the break point is to be considered an optional breakpoint, and thus similar to hyphenating words, or a mandatory break. The latter is the default. The tag is self-contained, i.e. it has no content.

```
<df>5x - 2y + 9 = x + <break> + 6y - 3 </df>
```

resulting in

$$\begin{aligned} &5x - 2y + 9 = x + \\ &+ 6y - 3 \end{aligned}$$

Super and subscripts <sup> </sup> pos=
<sub> </sub> pos=

The superscript and subscript tags are used to identify superscripts and subscripts in a formula. Both the start and end-tags are required. The position attribute (pos=) takes one of the values: pre, mid, or post, post being the default.

NOTE — These elements are purely for "layout"; squares and powers should be entered using the corresponding elements.

```
x<sub/2/
```

resulting in

$$x_2$$

and

```
x<sub/i/<sup/2/
```

resulting in

$$x_i^2$$

and, grouping with a fence,

```
<fence type=none>x<sub/2/</fence><sub/3/
```

resulting in

$${x_2}_3$$

and

```
lim<sub pos=mid/x &rarr; a/ f(x)
```

resulting in

$$\lim_{x \to a} f(x)$$

Boxes <box> </box> style=

The box tag is used to identify parts of a formula to be placed inside a box. Both the start and end-tags are required.

```
<box>a+b</box>
```

resulting in

$$\boxed{a+b}$$

'Over' embellishments <ov> </ov> type= pos= style=

The over-character tag is used to identify parts of a formula over which special accents or diacritical marks are to be placed. Both the start and end-tags are required. The type attribute (type=) is used to define the type of mark to be used. It takes one of the values: dot, dotdot, dot3, dot4, tie, tiebrace, hat, caron, acute, grave, cedil, ring, macron, ogonek, dblac, breve, tilde, vec, rvec, dyad, or bar. Bar is the default. The position attribute (pos=) defines the position of the mark and takes one of the values: above, below, or mid. Above is the default. The style attribute (style=) defines the style of the mark. It takes one of the values: single, double, triple, dash, dots, or bold. The default value is single.

NOTE — Not all combinations of type and style attribute values would normally be used; e.g. type=tilde style=dots.

```
<ov>x</ov>
<ov type=tilde>x</ov>
<ov pos=below>x+y</ov>
```

resulting in

$\overline{x}$
$\tilde{x}$
$\underline{x+y}$

Tensors <tensor> </tensor> suffix= posf=

The tensor tag is used to identify tensors in a formula. Both the start and end-tags are required. The suffix attribute (suffix=) takes the suffixes of the tensor; a space indicating a switch from superscript to subscript or subscript to superscript. The value of the position of the first suffix (posf=) attribute takes the value sup or sub, sup being the default.

```
<tensor suffix="i j k"/A/ = <tensor suffix="ip"/g/ <tensor suffix=kq/g/
<tensor posf=sub suffix="pjq"/A/
```

resulting in

$$A^{i\ k}_{\ j} = g^{ip} g^{kq} A_{pjq}$$

Functions <mfn> </mfn>
<fname>
<of>

By convention, most characters except numeral digits and syntax elements are set in italics in mathematical formulae. Exceptions to this convention include names of functions. The <mfn> tag is used to identify the function and its argument. Both the start and end-tags are required. An attribute is used to indicate the name of the function. Permitted values are and, antilog, arc, arccos, arcsin, arctan, arg, colog, cos, cosh, cot, coth, csc, ctn, deg, det, dim, exp, for, gcd, glb, hom, if, Im, ker, lg, lim, ln, log, lub, max, min mod, Re, sec, sin, sinh, tan, and tanh.

NOTE — Due to the restriction in the Reference Quantity Set limiting the number of tokens in a group to 32, two attributes have been used.

```
<mfn cos/x/
```

resulting in

cos(x)

or

cos x

depending on the style used.

Alternatively the function name can be identified by the use of an <fname> tag and the argument by an <of> tag.

```
<mfn><fname>myfunc<of>x+y</mfn>
```

resulting in

myfunc $(x+y)$

Roman and italic fonts <roman> </roman>
<italic> </italic>

For the rare cases where parts of a formula should be set in roman or italic, contrary to common practice and not being a function name — e.g. as part of some special name not previously known as a mathematical operation, the roman and italic tags can be used to indentify these parts.

```
abc <roman/text/ def
```

resulting in

*abc*text*def*

NOTE — Characters from other fonts, e.g. Fraktur, would be entered using entities.

Vectors <vec> </vec>

The vector tag is used to identify vectors in a formula. By convention these are set in bold roman or medium italic with an arrow above. Both the start and end-tags are required.

```
<vec/v/
```

resulting in

$\mathbf{v}$

or

$\vec{v}$

Fractions <frac> </frac> align=
<numer>
<over>

Fractions are identified by the fraction tag <frac>. The alignment attribute (align=) can take the values: left, right, and centre, specifying the alignment of the numerator and denominator. The default is to centre the expressions. The numerator tag <numer> is used to identify the numerator of the fraction. This tag can, and should, be omitted since the element is required. The denominator is identified by the <over> tag. The end-tag of this element can be omitted. This element is followed by the required </frac> tag.

```
<frac>a<over>b+c</frac>
```

resulting in

$$\frac{a}{b+c}$$

Derivatives <diff> </diff>
<diffof>
<by>

Derivatives are identified by the differential tag <diff>. The numerator tag <diffof> is used to identify the numerator of the derivative. This tag can, and should, be omitted since the element is required. The denominator is identified by the <by> tag. The end-tag of this element can be omitted. This element is followed by the required </diff> tag.

```
<diff>x<by>y</diff>
<diff type=partial>V<by>a</diff>
```

resulting in

$$\frac{dx}{dy} \qquad \frac{\partial V}{\partial a}$$

Sum, integral, product, and general plex ('limits')

<plex> </plex>
<operator>
<from>
<to>
<of>
<sum> </sum>
<integral> </integral>
<product> </product>

The <plex> tag is used to identify a general "limits" operator. The <plex> tag is immediately followed by an operator, optionally identified by the <operator> tag. This in turn is followed by from and/or to elements for the lower and upper limits, followed by an optional of element. This is then followed by a required </plex> tag.

```
<plex>&cup;<from>i=1<to>10<of>a<sub/i/</plex>
```

resulting in

$$\bigcup_{i=1}^{10} a_i$$

The sum, integral, and product elements are special cases of the general plex, the operator being implied by these tags.

```
<sum><from>i=1<to>10<of>a<sub/i/</sum>
```

resulting in

$$\sum_{i=1}^{10} a_i$$

Piles <pile> </pile> align=
<above1>
<above>

The <pile> tag is used to identify segments of a formula that are to be placed one above the other. The alignment attribute (align=) can take the values: left, right, and centre, specifying the alignment of the elements in the pile. The default is to centre the elements. The <pile> tag is immediately followed by an implied above1 element, where both the start and end-tags can and should be omitted. This is followed by one or more above elements, where the start-tag is required. The last is followed by the required </pile> tag.

```
<pile>a<above>b<above>c</pile>
```

resulting in

$$\begin{matrix} a \\ b \\ c \end{matrix}$$

Matrices

<matrix> </matrix>
<col> </col> align=
<above1>
<above>

The <matrix> tag is used to identify matrices. They are different from piles in that row elements of the matrix will always align horizontally rather than just occupy the required vertical space in the column. The <matrix> tag is immediately followed by a column <col> tag. The alignment attribute (align=) of the column tag can take the values: left, right, and centre, specifying the alignment of the elements in the column. The default is to centre the elements. The <col> tag is immediately followed by the above1 element, where both the start and end-tags can be omitted. This is followed by one or more above elements, where the start-tag is required. The last is followed by the required </col> tag. This can then be followed by further columns and the matrix is ended by the required </matrix> tag.

```
<matrix>
<col>1<above>2</col>
<col>0<above>3</col>
</matrix>
```

resulting in

$$\begin{matrix}1 & 0\\ 2 & 3\end{matrix}$$

Square root and square

<sqrt> </sqrt>
<square> </square>

Square roots are identified by the <sqrt> tag. Both the start and end-tags are required.

```
<sqrt>a+x</sqrt>
```

resulting in

$$\sqrt{a+x}$$

Squares are identified by the <square> tag. Both the start and end-tags are required.

```
<square>a+x</square>
```

resulting in

$$(a+x)^2$$

N-root and power

<root> </root>
<degree>
<of>
<power> </power>
<degree>
<of>

Roots of an arbitrary order are identified by the <root> tag. The start-tag is followed by the <degree> tag identifying the degree of the root, and by the <of> tag identifying the operand.

```
<root><degree>4<of>a+x</root>   <root><degree>a+b<of>x+y</root>
```

resulting in

$$\sqrt[4]{a+x} \quad \sqrt[a+b]{x+y}$$

NOTE — Both the start- and end-tags of the degree element can be omitted and the example can be keyboarded

```
<root>4<of>a+x</root>   <root>a+b<of>x+y</root>
```

Powers of an arbitrary order are identified by the <power> tag. The start-tag is followed by the <degree> tag identifying the degree of the power, and by the <of> tag identifying the operand.

```
<power><degree>4<of>a+x</power>   <power><degree>a+b<of>x+y</power>
```

resulting in

$$(a+x)^4 \quad (x+y)^{a+b}$$

NOTE — Both the start- and end-tags of the degree element can be omitted and the example can be keyboarded

```
<power>4<of>a+x</power>   <power>a+b<of>x+y</power>
```

Open and close brackets, fences, and posts

<fence> </fence>   type= style= open= close=
<middle> </middle>   style=

Fences are identified by the <fence> tag. The type attribute (type=) defines the character to be used for the fence. It takes one of the values: paren, bracket, angbrack, brace, bar, or none. Paren is the default. The style attribute (style=) defines the style of the fence. It takes one of the values: single, double, triple, dash, dots, or bold. The default value is single. Both the start and end-tags are required.

```
<fence><frac>a<over>b</frac></fence>
```

resulting in

$$\left(\frac{a}{b}\right)$$

```
<fence type=brace><frac>a<over>b</frac></fence>
```

resulting in

$$\left\{\frac{a}{b}\right\}$$

```
<fence type=bar style=double><frac>a<over>b</frac></fence>
```

resulting in

$$\left\|\frac{a}{b}\right\|$$

Unpaired fences, also single occurrences, are entered using the <fence> element and specifying the open and close attributes. The value of these attributes is the fence character to be used. Any value specified for the type attribute is ignored.

```
<fence open="(" close="]"><frac>a<over>b</frac> ,1 </fence>
```

resulting in

$$\left(\frac{a}{b},1\right]$$

A post is entered using the <middle> tag. The content of this element is the post character to be used. The style attribute (style=) defines the style of the post. It takes one of the values: single, double, triple, dash, dots, or bold. The default value is single.

```
<fence><frac>a<over>b</frac><middle/|/1</fence>
```

resulting in

$$\left(\frac{a}{b}\,\middle|\,1\right)$$

8.5 Short references

Five sets of short reference maps are used in this application: *fence* maps left parenthesis, square bracket, brace, and bar to the appropriate fence start-tag; four *infence* maps map the right (close) character corresponding to the active fence type to a fence end-tag and maps left parenthesis, square bracket, brace, and bar to the appropriate fence start-tag (of the nested fence).

The *fence* map is used for the inline and display formula element, thus being active also in contained elements. In the fence element one of the four *infence* maps is active.

NOTE — Bars, where open and close cannot be uniquely recognized, are restricted to pairs. Single occurences of parenthesis, bracket, brace, and bar is an error and has to be explicitly typed using the <fence> tag with an empty open or close attribute.

8.6 Complex examples

```
<df>a <sub/0/ +
<frac> b <sub/1/  <over> a <sub/1/ +
   <frac> b <sub/2/ <over> a <sub/2/ +
         <frac> b <sub/3/ <over> a <sub/3/ +
             <frac> b <sub/4/ <over> a <sub/4/ + ....
             </frac>
         </frac>
   </frac>
</frac></df>
```

resulting in

$$a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2 + \cfrac{b_3}{a_3 + \cfrac{b_4}{a_4 + \ldots.}}}}$$

```
<df>
c <sub/&mu;/ (t) =
<power> -<frac>i<over>h</frac>E<sub/&mu;/t <of> e </power>
 a <frac><fence><diff type=partial> V <by> a </diff></fence><sub/&mu;0/
  <over>
    E<sub/&mu;/-E<sub/0/
  </frac>
  <frac>
    <power><frac>i<over>h</frac>(E<sub/&mu;/-E<sub/0/) t <of>e</power> -1
  <over>
    <frac>i<over>h</frac> (E<sub/&mu;/-E<sub/0/)
  </frac>
</df>
```

resulting in

$$c_\mu(t) = e^{-\frac{i}{h}E_\mu t} a \frac{\left(\frac{\partial V}{\partial a}\right)_{\mu 0}}{E_\mu - E_0} \frac{e^{\frac{i}{h}(E_\mu - E_0)t} - 1}{\frac{i}{h}(E_\mu - E_0)}$$

8.7 Element and entity definition

```
<!-- (C) International Organization for Standardization 1988
     Permission to copy in any form is granted for use with
     conforming SGML systems and applications as defined in
     ISO 8879, provided this notice is included in all copies.
-->
<!-- element type definition for mathematical formula notation
          aimed at the author and thus minimization of keystrokes
          has not been of prime importance                            -->
<!--        Public entity sets described in Annex D
                of ISO 8879                                           -->
     <!ENTITY % ISOgrk1 PUBLIC
       "ISO 8879-1986//ENTITIES Greek Letters//EN">
     %ISOgrk1;
     <!ENTITY % ISOgrk2 PUBLIC
       "ISO 8879-1986//ENTITIES Monotoniko Greek//EN">
     %ISOgrk2;
     <!ENTITY % ISOnum PUBLIC
       "ISO 8879-1986//ENTITIES Numeric and Special Graphic//EN">
     %ISOnum;
     <!ENTITY % ISOpub PUBLIC
       "ISO 8879-1986//ENTITIES Publishing//EN">
     %ISOpub;
     <!ENTITY % ISOtech PUBLIC
       "ISO 8879-1986//ENTITIES General Technical//EN">
     %ISOtech;
     <!ENTITY % ISOgrk3 PUBLIC
       "ISO 8879-1986//ENTITIES Greek Symbols//EN">
     %ISOgrk3;
     <!ENTITY % ISOgrk4 PUBLIC
       "ISO 8879-1986//ENTITIES Alternative Greek Symbols//EN">
     %ISOgrk4;
     <!ENTITY % ISOamso PUBLIC
       "ISO 8879-1986//ENTITIES Added Math Symbols: Ordinary//EN">
     %ISOamso;
     <!ENTITY % ISOamsb PUBLIC
       "ISO 8879-1986//ENTITIES Added Math Symbols: Binary Operators//EN">
     %ISOamsb;
     <!ENTITY % ISOamsr PUBLIC
       "ISO 8879-1986//ENTITIES Added Math Symbols: Relations//EN">
     %ISOamsr;
     <!ENTITY % ISOamsn PUBLIC
       "ISO 8879-1986//ENTITIES
        Added Math Symbols: Negated Relations//EN">
     %ISOamsn;
     <!ENTITY % ISOamsa PUBLIC
       "ISO 8879-1986//ENTITIES Added Math Symbols: Arrow Relations//EN">
     %ISOamsa;
     <!ENTITY % ISOamsc PUBLIC
       "ISO 8879-1986//ENTITIES Added Math Symbols: Delimiters//EN">
     %ISOamsc;
<!--
                                                                      -->
<!ENTITY % f.oper  "mark|markref|break|sup|sub|roman|italic|
                    sum|integral|product|plex|
                    frac|diff|sqrt|root|square|power|
                    pile|matrix|fence|middle|tensor|mfn|
                    box|ov|vec"
                          -- all the operators                        -->
<!ENTITY % f.text  "#PCDATA|roman|italic|ov" -- "text"                -->
<!ENTITY % f.align "centre|left|right"  -- alignment of elements      -->
<!ENTITY % f.pos   "post|pre|mid"       -- position of elements       -->
<!ENTITY % f.style "single|double|triple|dash|dots|bold"
                                        -- style of elements          -->
<!ENTITY % f.type  "paren|bracket|angbrack|brace|bar|none"
                                                     -- fencetype     -->
<!ENTITY % f.ov    "dot|dotdot|dot3|dot4|tie|tiebrace|hat|haczeck|
  acute|grave|cedil|ring|macron|ogonek|dblac|
  breve|tilde|vec|rvec|dyad|bar"                       -- over chars  -->
<!ENTITY % f.diff  "normal|partial"     -- types of differentiation   -->
<!ENTITY % f.func1 "and|antilog|arc|arccos|arcsin|arctan|arg|
                    colog|cos|cosh|cot|coth|csc|ctn|deg|det|
                    dim|exp|for|gcd|glb|hom|if|Im|ker|lg|lim|
                    ln|log|lub|max|min"
                          -- 32 first functions                       -->
<!ENTITY % f.func2 "mod|Re|sec|sin|sinh|tan|tanh"
                          -- remaining functions                      -->
<!ENTITY   f.lpar  "<FENCE TYPE=paren><!USEMAP infencep>"               >
<!ENTITY   f.lsqu  "<FENCE TYPE=bracket><!USEMAP infences>"             >
<!ENTITY   f.lbra  "<FENCE TYPE=brace><!USEMAP infenceb>"               >
<!ENTITY   f.lbar  "<FENCE TYPE=bar><!USEMAP infencer>"                 >
<!ENTITY   f.rpar  "</FENCE>"                                           >
<!ENTITY   f.rsqu  "</FENCE>"                                           >
<!ENTITY   f.rbra  "</FENCE>"                                           >
<!ENTITY   f.rbar  "</FENCE>"                                           >
<!SHORTREF fence              -- Map for open brackets                 --
                  "("     f.lpar      -- left parenthesis     --
                  "["     f.lsqu      -- left square bracket  --
                  "{"     f.lbra      -- left brace           --
                  "|"     f.lbar      -- left bar             --
>
<!SHORTREF infencep           -- Map for open and close brackets       --
                  "("     f.lpar      -- left parenthesis     --
                  "["     f.lsqu      -- left square bracket  --
                  "{"     f.lbra      -- left brace           --
                  "|"     f.lbar      -- left bar             --
                  ")"     f.rpar      -- right parenthesis    --
>
<!SHORTREF infences           -- Map for open and close brackets       --
                  "("     f.lpar      -- left parenthesis     --
                  "["     f.lsqu      -- left square bracket  --
                  "{"     f.lbra      -- left brace           --
                  "|"     f.lbar      -- left bar             --
                  "]"     f.rsqu      -- right square bracket --
>
<!SHORTREF infenceb           -- Map for open and close brackets       --
                  "("     f.lpar      -- left parenthesis     --
                  "["     f.lsqu      -- left square bracket  --
                  "{"     f.lbra      -- left brace           --
                  "|"     f.lbar      -- left bar             --
                  "}"     f.rbra      -- right brace          --
>
<!SHORTREF infencer           -- Map for open and close brackets       --
                  "("     f.lpar      -- left parenthesis     --
                  "["     f.lsqu      -- left square bracket  --
                  "{"     f.lbra      -- left brace           --
                  "|"     f.rbar      -- right bar            --
>
<!--
       inline formula
             Example:  <f><frac>a<over>b</frac></f>
                                                                     -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                      -->
<!ELEMENT f           - -  (%f.text;|%f.oper;)+                        >
<!USEMAP  fence       f                                                >
<!--
       display formula
             Example:  <df><frac>a<over>b</frac></df>
                                                                     -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                      -->
<!ELEMENT df          - -  (%f.text;|%f.oper;)+                        >
<!--      ELEMENTS    NAME     VALUE        DEFAULT                  -->
<!ATTLIST df          id       ID           #IMPLIED
                      align    (%f.align)   left
                      num      CDATA        #IMPLIED                   >
<!USEMAP  fence       df                                               >
<!--
       display formula group
             Example:  <dfg>
                       <df><frac>a<over>b</frac></df>
                       <df><frac>c<over>d</frac></df>
                       </dfg>
                                                                     -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                      -->
<!ELEMENT dfg         - -  (df)+                                       >
<!--      ELEMENTS    NAME     VALUE        DEFAULT                  -->
<!ATTLIST dfg         id       ID           #IMPLIED
                      align    (%f.align)   left
                      num      CDATA        #IMPLIED                   >
<!--
       formula reference
             Example:  <dfref refid=x>
                                                                     -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                      -->
<!ELEMENT dfref       - O  EMPTY                                       >
<!--      ELEMENTS    NAME     VALUE        DEFAULT                  -->
<!ATTLIST dfref       page     (yes|no)     yes
                      refid    IDREF        #REQUIRED                  >
<!--
       mark and markref - for horizontal alignment
       break - for division points in formula
             Example:  <mark id=x>
                       <markref refid=x>
                                                                     -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                      -->
<!ELEMENT mark        - O  EMPTY                                       >
<!--      ELEMENTS    NAME     VALUE        DEFAULT                  -->
<!ATTLIST mark        id       ID           #REQUIRED                  >
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                      -->
<!ELEMENT markref     - O  EMPTY                                       >
<!--      ELEMENTS    NAME     VALUE        DEFAULT                  -->
<!ATTLIST markref     refid    IDREF        #REQUIRED                  >
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                      -->
<!ELEMENT break       - O  EMPTY                                       >
<!--      ELEMENTS    NAME     VALUE                 DEFAULT         -->
<!ATTLIST break       type     (required|optional)   required          >
<!--
       boxes
```

```
             Example:  <BOX>....</BOX>
                                                                      -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                       -->
<!ELEMENT box         - -   (%f.text;|%f.oper;)+                        >
<!--      ELEMENTS    NAME     VALUE       DEFAULT                    -->
<!ATTLIST box         style    (%f.style;) single                       >
<!--
       "over" embellishments
             Example:  <OV>....</OV>
                                                                      -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                       -->
<!ELEMENT ov          - -   (%f.text;|%f.oper;)+                        >
<!--      ELEMENTS    NAME     VALUE              DEFAULT             -->
<!ATTLIST ov          type     (%f.ov;)           bar
                      pos      (above|below|mid)  above
                      style    (%f.style;)        single                >
<!--
       super and subscripts
             Example:  x<sup/2/
                                                                      -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                       -->
<!ELEMENT (sup|sub)   - -   (%f.text;|%f.oper;)+                        >
<!--      ELEMENTS    NAME     VALUE       DEFAULT                    -->
<!ATTLIST (sup|sub)   pos      (%f.pos;)   post                         >
<!--
       tensors
             Example:  <tensor suffix="i j k"/A/
                                                                      -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                       -->
<!ELEMENT tensor      - -   (#PCDATA)                                   >
<!--      ELEMENTS    NAME     VALUE       DEFAULT                    -->
<!ATTLIST tensor      posf     (sup|sub)   sup
                      suffix   CDATA       #REQUIRED                    >
<!--
       roman and italic fonts
             Example:  <roman/text/
                                                                      -->
<!--      ELEMENTS        MIN  CONTENT (EXCEPTIONS)                   -->
<!ELEMENT (roman|italic)  - -  (#PCDATA)                                >
<!--
       vectors
             Example:  <vec/v/
                                                                      -->
<!--      ELEMENTS        MIN  CONTENT (EXCEPTIONS)                   -->
<!ELEMENT vec             - -  (#PCDATA)                                >
<!--
       sum, integral, product, and general plex ("limits")
             Example:  <sum><from>i=1<to>10<of>a<sub/i/</sum>
                       <plex>&cup;<from>i=1<to>10<of>a<sub/i/</plex>
                                                                      -->
<!--      ELEMENTS      MIN  CONTENT (EXCEPTIONS)                     -->
<!ELEMENT sum           - -  ((from|to)*,of?)                           >
<!ELEMENT integral      - -  ((from|to)*,of?)                           >
<!ELEMENT product       - -  ((from|to)*,of?)                           >
<!ELEMENT plex          - -  (operator,(from|to)*,of?)                  >
<!ELEMENT (from|to|of)  - O  (%f.text;|%f.oper;)+                       >
<!ELEMENT operator      O O  (%f.text;)                                 >
<!--
       fractions
             Example:  <frac>a<over>b+c</frac>
                                                                      -->
<!--      ELEMENTS    MIN  CONTENT (EXCEPTIONS)                       -->
```

```
<!ELEMENT frac         - -  (numer,over)                            >
<!--      ELEMENTS     NAME     VALUE       DEFAULT              -->
<!ATTLIST frac         align    (%f.align;) centre                  >
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT numer        0 0  (%f.text;|%f.oper;)+                    >
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT over         - 0  (%f.text;|%f.oper;)+                    >
<!--
       differentials
             Example:  <diff>x<by>y</diff>
                                                                 -->
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT diff         - -   (diffof,by)                            >
<!--      ELEMENTS     NAME     VALUE       DEFAULT              -->
<!ATTLIST diff         type     (%f.diff;)  normal                  >
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT diffof       0 0  (%f.text;|%f.oper;)+                    >
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT by           - 0  (%f.text;|%f.oper;)+                    >
<!--
       piles
             Example:  <pile>a<above>b<above>c</pile>
                                                                 -->
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT pile         - -   (above1,(above)+)                      >
<!--      ELEMENTS     NAME     VALUE       DEFAULT              -->
<!ATTLIST pile         align    (%f.align;) centre                  >
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT above1       0 0  (%f.text;|%f.oper;)+                    >
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT above        - 0  (%f.text;|%f.oper;)+                    >
<!--
       matrices
             Example:  <matrix>
                       <col>1<above>2</col>
                       <col>0<above>3</col>
                       </matrix>
                                                                 -->
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT matrix       - -  ((col)+)                                >
<!ELEMENT col          - -  (above1,(above)+)                       >
<!--      ELEMENTS     NAME     VALUE       DEFAULT              -->
<!ATTLIST col          align    (%f.align;) centre                  >
<!--
       square root and n-root
             Example:  <sqrt>a+x</sqrt>
                       <root><degree>4<of>a+x</root>
                                                                 -->
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT sqrt         - -  (%f.text;|%f.oper;)+                    >
<!ELEMENT root         - -  (degree,of)                             >
<!ELEMENT degree       0 0  (%f.text;|%f.oper;)+                    >
<!--
       square and power
             Example:  <square>a+x</square>
                       <power><degree>4<of>a+x</power>
                                                                 -->
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                 -->
<!ELEMENT square       - -  (%f.text;|%f.oper;)+                    >
<!ELEMENT power        - -  (degree,of)                             >
<!--
       mathematical functions
```

```
             Example:  <mfn ln>1+x</mfn>
                       <mfn cos/&pi;/
                       <mfn><fname>myfunc<of>x+y</mfn>
                                                                      -->
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                      -->
<!ELEMENT mfn          - -  ((fname,of)|(%f.text;|%f.oper;)+)           >
<!--      ELEMENTS     NAME     VALUE        DEFAULT                  -->
<!ATTLIST mfn          type1    (%f.func1;) #IMPLIED
                       type2    (%f.func2;) #IMPLIED                    >
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                      -->
<!ELEMENT fname        - O  (#PCDATA)                                   >
<!--
       fences and posts, open and close brackets
             Example: <fence>a+b</fence>
                      <fence open="(" close="]">a+b</fence>
                                                                      -->
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                      -->
<!ELEMENT fence        - -  (%f.text;|%f.oper;)+                        >
<!--      ELEMENTS     NAME     VALUE        DEFAULT                  -->
<!ATTLIST fence        type     (%f.type;)   paren
                       open     CDATA        #IMPLIED
                       close    CDATA        #IMPLIED
                       style    (%f.style;)  single                     >
<!USEMAP  fence        fence                                            >
<!--      ELEMENTS     MIN  CONTENT (EXCEPTIONS)                      -->
<!ELEMENT middle       - -  (%f.text;)                                  >
<!--      ELEMENTS     NAME     VALUE        DEFAULT                  -->
<!ATTLIST middle       style    (%f.style;)  single                     >
<!USEMAP  #EMPTY       middle                                           >
```

9 Tables

9.1 Table elements

table <table> </table> id=

The table tag <table> precedes a table, an end-tag being required. The value of the identifier attribute (id=) can be used when referring to the table and must be a unique reference in the document. Associated with a table may be a table caption and a table description. <table> is followed immediately by another tag. The end-tag </table> is input after all the constituent elements for the table.

table heading level 1-4 <th1>, <th2>, <th3>, <th4>

These tags precede any headings of a table, <th1> being the first tag input after <table>. If this heading is subdivided, then its constituent subheadings are input next; if any subheading is further subdivided, then its constituent subheadings are input before going to the next subheading (or heading). No end-tag is necessary. Valid only in a table element.

row (of table) <r>

The <r>tag precedes any entries in a row of a table. No end-tag is necessary. Valid only in a table element. <r> is followed immediately by a <c> tag.

column (of table) <c>

The column tag <c> precedes any entry for a column in a table. No end-tag is necessary. Valid only in a table element.

table caption <tabcap>

The table caption is preceded by <tabcap>. No end-tag is necessary. Valid only in a table element.

table description <tabdesc>

The description of a table is preceded by <tabdesc>. No end-tag is necessary. Valid only in a table element.

table reference <tabref> refid= page=

The table reference is used within the document to refer to an identified table, being replaced by the table number and caption (if there is one). If the page attribute is present, (page=yes), the appropriate page number is supplied by the system. The appropriate value for the reference identifier attribute (refid=) must be provided. The tag is self-contained, that is it has no content.

```
For further details see <tabref refid=arfhc page=yes>.
```

9.2 Element definiton

```
<!--      ELEMENTS  MIN  CONTENT (EXCEPTIONS) -->
<!ELEMENT table     - -  (th1*,r*,tabcap,tabdesc?) -- table -->
<!ELEMENT th1       - O  ((#PCDATA|th2)*) -- table heading level 1 -->
<!ELEMENT th2       - O  ((#PCDATA|th3)*) -- table heading level 2 -->
<!ELEMENT th3       - O  ((#PCDATA|th4)*) -- table heading level 3 -->
<!ELEMENT th4       - O  (#PCDATA)        -- table heading level 4 -->
<!ELEMENT r         - O  (c*)         -- table row -->
<!ELEMENT c         - O  (#PCDATA)    -- table column -->
<!ELEMENT (tabcap|tabdesc) - O (#PCDATA) -- caption or description -->
<!ELEMENT tabref    - O  EMPTY        -- table reference -->
<!--      ELEMENTS  NAME     VALUE    DEFAULT -->
<!ATTLIST table     id       ID       #IMPLIED>
<!ATTLIST tabref    refid    IDREF    #REQUIRED>
```

9.3 Example of a marked-up table

```
<table id=arfhc>
<th1>Year
<th1>Annual Rainfall (in centimetres)
<th2>First Quarter  <th3>Jan<th3>Feb<th3>Mar
<th2>Second Quarter <th3>Apr<th3>May<th3>Jun
<th2>Third Quarter  <th3>Jul<th3>Aug<th3>Sep
<th2>Fourth Quarter <th3>Oct<th3>Nov<th3>Dec
<th1>Total
<r><c>1986<c>3.8<c>3.6<c>4.4<c>5.0<c>6.0<c>6.2
<c>5.8<c>7.2<c>7.2<c>5.6<c>6.2<c>4.8<c>65.8
<r><c>1987<c> ...
<tabcap>Annual Rainfall for Head City
</table>
```

resulting in