International Standard 8879

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION•МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ•ORGANISATION INTERNATIONALE DE NORMALISATION

Information processing — Text and office systems — Standard Generalized Markup Language (SGML)

Traitement de l'information — Systèmes bureautiques — Langage standard généralisé de balisage (SGML)

First edition — 1986-10-15

UDC 681.3.06 **Ref. No. ISO 8879-1986 (E)**

Descriptors : data processing, documentation, logical structure, programming (computers), artificial languages, programming languages.

Price based on 155 pages

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75 % approval by the member bodies voting.

International Standard ISO 8879 was prepared by Technical Committee ISO/TC 97, *Information processing systems*.

Users should note that all International Standards undergo revision from time to time and that any reference made herein to any other International Standard implies its latest edition, unless otherwise stated.

Printed in Switzerland

Contents

Annexes

Figures

This page intentionally left blank

Information Processing — Text and Office Systems — Standard Generalized Markup Language (SGML)

0 Introduction

This International Standard specifies a language for document representation referred to as the "Standard Generalized Markup Language" (SGML). SGML can be used for publishing in its broadest definition, ranging from single medium conventional publishing to multi-media data base publishing. SGML can also be used in office document processing when the benefits of human readability and interchange with publishing systems are required.

0.1 Background

A document can be viewed in the abstract as a structure of various types of element. An author organizes a book into chapters that contain paragraphs, for example, and figures that contain figure captions. An editor organizes a magazine into articles that contain paragraphs that contain words, and so on.

Processors treat these elements in different ways. A formatting program might print headings in a prominent type face, leave space between paragraphs, and otherwise visually convey the structure and other attributes to the reader. An information retrieval system would perhaps assign extra significance to words in a heading when creating its dictionary.

Although this connection between a document's attributes and its processing now seems obvious, it tended to be obscured by early text processing methods. In the days before automated typesetting, an editor would "mark up" a manuscript with the specific processing instructions that would create the desired format when executed by a compositor. Any connection between the instructions and the document's structure was purely in the editor's head.

Early computerized systems continued this approach by adding the process-specific "markup" to the machine-readable document file. The markup still consisted of specific processing instructions, but now they were in the language of a formatting program, rather than a human compositor. The file could not easily be used for a different purpose, or on a different computer system, without changing all the markup.

As users became more sophisticated, and as text processors became more powerful, approaches were developed that alleviated this problem. "Macro calls" (or "format calls") were used to identify points in the document where processing was to occur. The actual processing instructions were kept outside of the document, in "procedures" (or "macro definitions" or "stored formats"), where they could more easily be changed.

While the macro calls could be placed anywhere in a document, users began to recognize that most were placed at the start or end of document elements. It was natural, therefore, to choose names for such macros that were "generic identifiers" of the element types, rather than names that suggested particular processing (for example, "heading" rather than "format-17"), and so the practice of "generic coding" (or "generalized tagging") began.

Generic coding was a major step towards making automated text processing systems reflect the natural relationship between document attributes and processing. The advent of "generalized markup languages" in the early 1970's carried this trend further by providing a formal language base for generic coding. A generalized markup language observes two main principles:

a) Descriptive markup predominates and is distinguished from processing instructions.

   Descriptive markup includes both generic identifiers and other attributes of document elements that motivate processing instructions. The processing instructions, which can be in any language, are normally collected outside of the document in procedures.

   As the source file is scanned for markup and the various elements are recognized, the processing system executes the procedures associated with each element and attribute for that process. For other processes, different procedures can be associated with the same elements and attributes without changing the document markup.

   When a processing instruction must be entered directly in a document, it is delimited differently from descriptive markup so that it can easily be located and changed for different processes.

b) Markup is formally defined for each type of document.

   A generalized markup language formalizes document markup by incorporating "document type definitions". Type definitions include a specification (like a formal grammar) of which elements and attributes can occur in a document and in what order. With this information it is possible to determine whether the markup for an individual document is correct (that is, complies with the type definition) and also to supply markup that is missing, because it can be inferred unambiguously from other markup that is present.

NOTE — A more detailed introduction to the concepts of generic coding and the Standard Generalized Markup Language can be found in annex A.

0.2 Objectives

The Standard Generalized Markup Language standardizes the application of the generic coding and generalized markup concepts. It provides a coherent and unambiguous syntax for describing whatever a user chooses to identify within a document. The language includes:

— An "abstract syntax" for descriptive markup of document elements.

— A "reference concrete syntax" that binds the abstract syntax to particular delimiter characters and quantities. Users can define alternative concrete syntaxes to meet their requirements.

— Markup declarations that allow the user to define a specific vocabulary of generic identifiers and attributes for different document types.

— Provision for arbitrary data content. In generalized markup, "data" is anything that is not defined by the markup language. This can include specialized "data content notations" that require interpretation different from general text: formulas, images, non-Latin alphabets, previously formatted text, or graphics.

— Entity references: a non-system-specific technique for referring to content located outside the mainstream of the document, such as separately-written chapters, pi characters, photographs, etc.

— Special delimiters for processing instructions to distinguish them from descriptive markup. Processing instructions can be entered when needed for situations that cannot be handled by the procedures, but they can easily be found and modified later when a document is sent to a different processing system.

For a generalized markup language to be an acceptable standard, however, requires more than just providing the required functional capabilities. The language must have metalinguistic properties, in order to satisfy the constraints imposed by the need to use it in a multiplicity of environments. The major constraints, and the means by which the Standard Generalized Markup Language addresses them, can be summarized as follows:

a) Documents "marked up" with the language must be processable by a wide range of text processing and word processing systems.

   The full form of the language, with all optional features, offers generality and flexibility that can be exploited by sophisticated systems; less powerful systems need not support the features. To facilitate interchange between dissimilar systems, an "SGML declaration" describes any markup features or concrete syntax variations used in a document.

b) The millions of existing text entry devices must be supported.

   SGML documents, with the reference concrete syntax, can easily be keyboarded and understood by humans, without machine assistance. As a result:

— Use of SGML need not await the development and acceptance of a new generation of hardware — just software to process the documents on existing machines.
— Migration to such a new generation (when it comes) will be easier, as users will already be familiar with SGML.

c) There must be no character set dependency, as documents might be keyed on a variety of devices.

The language has no dependency on a particular character set. Any character set that has bit combinations for letters, numerals, space, and delimiters is acceptable.

d) There must be no processing, system, or device dependencies.

Generalized markup is predominantly descriptive and therefore inherently free of such dependencies. The occasional processing instruction is specially delimited so it can be found and converted for interchange, or when a different process renders the instruction irrelevant.

References to external parts of a document are indirect. The mappings to real system storage are made in "external entity declarations" that occur at the start of the document, where they can easily be modified for interchange.

The concrete syntax can be changed with the SGML declaration to accommodate any reserved system characters.

e) There must be no national language bias.

The characters used for names can be augmented by any special national characters. Generic identifiers, attribute names, and other names used in descriptive markup are defined by the user in element and entity declarations.

The declaration names and keywords used in markup declarations can also be changed.

Multiple character repertoires, as used in multi-lingual documents, are supported.

f) The language must accommodate familiar typewriter and word processor conventions.

The "short reference" and "data tag" capabilities support typewriter text entry conventions. Normal text containing paragraphs and quotations is interpretable as SGML although it is keyable with no visible markup.

g) The language must not depend on a particular data stream or physical file organization.

The markup language has a virtual storage model in which documents consist of one or more storage entities, each of which is a sequence of characters. All real file access is handled by the processing system, which can decide whether the character sequence should be viewed as continuous, or whether it should reflect physical record boundaries.

h) "Marked up" text must coexist with other data.

A processing system can allow text that conforms to this International Standard to occur in a data stream with other material, as long as the system can locate the start and end of the conforming text.

Similarly, a system can allow data content not defined by SGML to occur logically within a conforming document. The occurrence of such data is indicated by markup declarations to facilitate interchange.

i) The markup must be usable by both humans and programs.

The Standard Generalized Markup Language is intended as a suitable interface for keyboarding and interchange without preprocessors. It allows extensive tailoring to accommodate user preferences in text entry conventions and the requirements of a variety of keyboards and displays.

However, it is recognized that many implementers will want to take advantage of the language's information capture capabilities to provide intelligent editing or to create SGML documents from a word processing front-end environment. SGML accommodates such uses by providing the following capabilities:
— Element content can be stored separately from the markup.
— Control characters can be used as delimiters.
— Mixed modes of data representation are permitted in a document.
— Multiple concurrent logical and layout structures are supported.

0.3 Organization

The organization of this International Standard is as follows:

a) The physical organization of an SGML document as an entity structure is specified in clause 6.

b) The logical organization of an SGML document as an element structure, and its representation with descriptive markup, is specified in clause 7.

c) Processing instructions are discussed in clause 8.

d) Common markup constructs, such as characters, entity references, and processing instructions, are covered in clause 9.

e) Markup declarations with general applicability (comment, entity, and marked section) are specified in clause 10.

f) Markup declarations that are used primarily to specify document type definitions (document type, element, notation, short reference mapping, and short reference use) are defined in clause 11.

g) Markup declarations that are used primarily to specify link process definitions (link type, link attribute, link set, and link set use) are defined in clause 12.

h) The SGML declaration, which specifies the document character set, capacity set, concrete syntax, and features, is defined in clause 13.

i) The reference concrete syntax is defined in clause 14.

j) Conformance of documents, applications, and systems is defined in clause 15.

There are also a number of annexes containing additional information; they are not integral parts of the body of this International Standard.

NOTE — This International Standard is a formal specification of a computer language, which may prove difficult reading for those whose expertise is in the production of documents, rather than compilers. Annexes A, B, and C discuss the main concepts in an informal tutorial style that should be more accessible to most readers. However, the reader should be aware that those annexes do not cover all SGML constructs, nor all details of those covered, and subtle distinctions are frequently ignored in the interest of presenting a clear overview.

1 Scope

This International Standard:

a) Specifies an abstract syntax known as the Standard Generalized Markup Language (SGML). The language expresses the description of a document's structure and other attributes, as well as other information that makes the markup interpretable.

b) Specifies a reference concrete syntax that binds the abstract syntax to specific characters and numeric values, and criteria for defining variant concrete syntaxes.

c) Defines conforming documents in terms of their use of components of the language.

d) Defines conforming systems in terms of their ability to process conforming documents and to recognize markup errors in them.

e) Specifies how data not defined by this International Standard (such as images, graphics, or formatted text) can be included in a conforming document.

NOTE — This International Standard does not:

a) Identify or specify "standard" document types, document architectures, or text structures.

b) Specify the implementation, architecture, or markup error handling of conforming systems.

c) Specify how conforming documents are to be created.

d) Specify the data stream, message handling system, file structure, or other physical representation in which conforming documents are stored or interchanged, or any character set or coding scheme into or from which conforming documents might be translated for such purposes.

e) Specify the data content representation or notation for images, graphics, formatted text, etc., that are included in a conforming document.

2 Field of Application

The Standard Generalized Markup Language can be used for documents that are processed by any text processing or word processing system. It is particularly applicable to:

a) Documents that are interchanged among systems with differing text processing languages.

b) Documents that are processed in more than one way, even when the procedures use the same text processing language.

Documents that exist solely in final formatted form are not within the field of application of this International Standard.

3 References

ISO 639, *Codes for the representation of names of languages.*[1)]

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

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.*[1)]

The following references are used in conjunction with illustrative material:

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

ISO 3166, *Codes for the representation of names of countries.*

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.*[1)]

ISO 8632/4, *Information processing systems — Computer graphics — Metafile for the storage and transfer of picture description information — Part 4: Clear text encoding.*[1)]

---

1) At present at the stage of draft.