Second edition
1997-08-01

Information technology — Hypermedia/Time-based Structuring Language (HyTime)

Technologies de l'information — Langage de structuration temporelle/hypermédia (HyTime)

Reference number
ISO/IEC 10744:1997(E)

Contents

ISO/IEC Copyright Office • Case Postale 56 • CH-1211 • Genève 20 • Switzerland

Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. 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. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

International Standard ISO/IEC 10744 was prepared by Joint Technical Committee JTC1, *Information technology*, Subcommittee SC 18, *Document processing and related communication*.

This second edition cancels and replaces the first edition (ISO/IEC 10744:1992), which has been technically revised.

Annexes A, B, and C form an integral part of this International Standard. Annex D is for information only.

Introduction

The Hypermedia/Time-based Structuring Language (HyTime), defined in this International Standard, provides facilities for representing static and dynamic information that is processed and interchanged by hypertext and multimedia applications. HyTime is an application of ISO 8879, the Standard Generalized Markup Language (SGML).

HyTime supports the classic bibliographic model of information referencing, whereby it is possible to represent links to anything, anywhere, at any time, in a variety of ways. The extension of this model to the computerized information age, known as "integrated open hypermedia" (IOH), is the field of application of HyTime.

HyTime provides standardized mechanisms for specifying interconnections (hyperlinks) within and between documents and other information objects, and for scheduling multimedia information in time and space.

Without HyTime, such information is typically embedded in the processing instructions of hypermedia "scripts" that govern the rendition of such documents, and is therefore not usable for other forms of processing. When HyTime is used, those properties of the information that are independent of specific processing are available for processing by applications and platforms other than the one on which the information was created.

It is for the application designer and user to decide which properties can be isolated from the scripts in this way. In an ideal world, the sole consideration would be whether the properties are intrinsic to the information, regardless of how it is processed. For example, the title of this clause is intrinsic information; the font that it appears in normally is not.

In the real world, representation strategies will vary from one situation to another and will depend on such other considerations as the expected uses of the information, the flexibility of the scripting language, and performance considerations. For this reason, HyTime is highly modularized so that application designers need use only the facilities for the properties they care to describe in a standardized way.

HyTime's rules for the standardized expression of hypermedia structuring are expressed as an "enabling architecture", consisting of a number of "architectural forms" and their associated semantics. The HyTime standard's formal definition as an architecture conforms to the Architectural Form Definition Requirements in annex A of this International Standard.

0.1 HyTime modules

The architectural forms and attributes of the HyTime language are grouped into five modules, each of which have both required and optional facilities. Support for the modules and their options is indicated by "HyTime support declarations."

— Base module

The base module consists of independent utility facilities, some of which are optional. The required facilities support hyperdocument management (using SGML) and identification of object properties. The optional facilities provide lookup tables for commonly used elements, a mechanism for associating use and access policies with objects, and a mechanism for relating attributes and the content of elements to their semantic values by reference. The base module also defines the fundamental coordinate addressing notation used by all the other HyTime modules.

— Location address module

The location address module allows the identification of objects that cannot be addressed by SGML unique identifiers, and objects that are in external documents.

Three basic types of address may be supported: name, semantic location, and coordinate location. Addressing of multiple locations is also possible. The syntax and semantics of these addressing mechanisms are independent of the data content notations of the data being addressed.

NOTE 1 The ability to resolve HyTime addresses in a given notation is dependent on software that can interpret that notation in terms of the abstractions HyTime uses for all addressing (see *6.1.1 Object representation*).

HyTime's system- and notation-independent way of expressing addresses of hypermedia objects also provides the basis of its hyperlinking and scheduling power.

— Hyperlinks module

This module allows relationships ("hyperlinks") to be established among objects, either within a single document or among the constituent documents and information objects of a hyperdocument.

— Scheduling module

This module allows events — occurrences of objects — to be scheduled on the coordinate axes of "finite coordinate spaces" in such a way that their positions can be expressed in terms of their relationships to one another. Measurement along the coordinate axes can be in terms of spatial or temporal units.

— Rendition module

When the scheduling module is used, object modification and/or event projection can be used to represent parameters governing the rendition process.

- Object modification

  The object modification facility allows specification of the order in which objects are to be modified during rendition and of the "object modifiers" (such as amplifiers and filters) that will affect them.

  NOTE 2 The semantics of the modifiers are not defined by HyTime.

- Event projection

  Rendition requires the projection of events into a coordinate space where they can be perceived; for example, from a coordinate space with a virtual time axis to one based on real time. The event projection facility allows specification of the factors for computing the positions and sizes of events in the target coordinate space.

  In situations where the rendered position and size of an event is not predictable (as when user interaction will affect it), the virtual dimensions of the original events may be projected onto real space/time via a formula in some arbitrary user-defined expression language. Such an expression may, among other things, accept late-binding values during rendition to resolve the positions and sizes of projected events.

  NOTE 3 The semantics of formatting the objects to fit the new extents is not defined by HyTime.

Applications can choose to include rendition information as an essential part of a hyperdocument, or it can be incorporated in the "style sheets" of the processing programs. The choice depends on the nature of the information being rendered. In multimedia documents, for example, rendition style tends to be more essential to the document than is the case in conventional documents.

0.2 HyTime applications

HyTime provides a generic level of support for a variety of applications, rather than the semantics for a specific application (that is, HyTime is like a carrier or infrastructure).

The boundary between an application and HyTime is variable, and is determined by the application designer, who is free to decide how much of the information will be expressed in a standardized way using HyTime and how much will be application-specific (for example, in a data content notation).

Because the semantics of HyTime's architectural forms and attributes are standardized, it will be possible to implement supporting software and/or hardware usable for a variety of applications. Applications can define additional attributes when defining an element type that is based on an architectural form. The semantics of the application-defined element types and attributes are the only ones defined by the applications themselves. They could be standardized by an industry group or formally by a national or international standards body.

HyTime attributes have no intrinsic meaning other than that specified in this International Standard. However, an application can impute additional semantics to them, either implicitly, or by defining appropriate element types and attributes. For example, to HyTime, the "dimension reference" architectural form means only that the dimension of one object is calculated from the dimension of another. An application, however, could specify (if it wished) that use of dimension referencing implies a synchronization relationship between the objects, and could emphasize this by using "sync" as the generic identifier of a dimension reference element type.

HyTime elements can occur wherever an application's DTD and the HyTime meta-DTD allow. A finite coordinate space could occur, for example, within a paragraph of a memo in order to represent a calendar or project plan in that context, or several paragraphs could occur as the content of a timed event.

Clients of HyTime, including applications and application architectures, can define non-HyTime architectural forms as well as elements. Although an application may not add architectural forms to HyTime, nor combine HyTime architectural forms with one another, it can create its own architecture (for example, "MyArch") defining its own set of architectural forms. These architectures may be derived wholly or in part from the HyTime architecture. The facilities for defining and using architectures are defined in annex A.3.

If, for example, a document is derived from the HyTime and MyArch architectures, after the content and attributes of each element are processed and validated in SGML terms by the SGML parser, elements with HyTime attributes would be subject to processing and validation by the HyTime engine, while elements with MyArch attributes would be subject to appropriate processing and validation by the application, perhaps aided by a MyArch engine.

HyTime defines some of the parameters needed by an application to accomplish rendition, and some of the rendition functionality. The remainder is provided by the application, or by a document architecture to which the application conforms.

Many different HyTime-conforming applications and architectures could exist, to address different requirements and serve different user constituencies. Such architectures could be incompatible in their non-HyTime aspects, but would still be supportable by a single HyTime engine.

NOTE 4 For example, no application would need to invent its own system for representing finite coordinate spaces, even if its projection functions were extremely intricate and application-specific. HyTime allows application-specific projection functions, using application-chosen (or defined) function languages, to be represented in conjunction with standardized representations of the unprojected and projected finite coordinate spaces.

HyTime's design is optimized for the sequencing and alignment problems encountered in typical hypermedia applications; it is not intended as a