INTERNATIONAL STANDARD

ISO/IEC 14478-1

First edition
1998-12-15

Information technology — Computer graphics and image processing — Presentation Environment for Multimedia Objects (PREMO) —

Part 1:
Fundamentals of PREMO

Technologies de l'information — Infographie et traitement d'images — Environnement de présentation d'objets multimédia (PREMO) —

Partie 1: Principes fondamentaux de PREMO

Reference number
ISO/IEC 14478-1:1998(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, government 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 JTC1. Draft International Standards adopted by the joint technical committees are circulated to the national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

ISO/IEC 14478–1 was prepared by Joint Technical Committee ISO/IEC JTC1, *Information technology*, Subcommittee SC24, *Computer graphics and image processing*.

ISO/IEC 14478consists of the following parts under the general title *Information technology — Computer graphics and image processing — Presentation Environment for Multimedia Objects (PREMO):*

- *Part 1: Fundamentals of PREMO*
- *Part 2: Foundation Component*
- *Part 3: Multimedia Systems Services*
- *Part 4: Modelling, Rendering, and Interaction Component*

Additional parts may be defined as this work progresses.

Annexes A and B form an integral part of this part of ISO/IEC 14478. Annex C is for information only.

Introduction

The need for a coordinated method for addressing all aspects of the construction of, presentation of, and interaction with multimedia objects has led to the standardization of this Presentation Environment for Multimedia Objects. Multimedia means objects consisting of still computer graphics, moving computer graphics (animation), synthetic graphics of all types, audio, still images, moving images (including video), images coming from imaging operations, and any other content type or combination of content types that can be "presented". ISO/IEC 14478 is extensible and configurable, and allows the separate, incremental development of additional standardized and non–standardized components to meet the needs of application communities.

PREMO currently consists of the following parts:

Part 1: Fundamentals of PREMO

Contains a motivational overview of PREMO giving its scope, justification, and an explanation of key concepts, describes the overall architecture of PREMO, and specifies the common semantics for specifying the externally visible characteristics of PREMO objects in an implementation-independent way.

Part 2: Foundation component

This component lists an initial set of object types and non–object types useful for the construction of, presentation of, and interaction with multimedia information. Any conforming PREMO implementation shall support these object types.

Part 3: Multimedia Systems Services Component

Describes objects that provide an infrastructure for building multimedia computing platforms that support interactive multimedia applications dealing with synchronized, time-based media in a heterogeneous distributed environment.

Part 4: Modelling, Presentation, and Interaction Component

Describes objects which are needed for advanced computer systems using graphics, video, audio, or other types of presentable media enhanced by time aspects.

NOTE — Further internationally standardized components are expected to be developed within ISO/IEC JTC1/SC24 and by other subcommittees.

Information technology — Computer graphics and image processing — Presentation Environment for Multimedia Objects (PREMO) — Part 1: Fundamentals of PREMO

1 Scope

ISO/IEC 14478 specifies techniques for supporting interactive single, and multiple media applications which recognize and emphasize the interrelationships among user interfaces, multimedia applications, and multimedia information interchange.

ISO/IEC 14478 defines a flexible environment to encompass modular functionality and is extensible through the creation of future components, both within and outside of standards committees. It supports a wide range of multimedia applications in a consistent way, from simple drawings up to full motion video, sound, and virtual reality environments.

ISO/IEC 14478 is independent of any particular implementation language, development environment, or execution environment. For integration into a programming environment, the standard shall be embedded in a system dependent interface following the particular conventions of that environment. ISO/IEC 14478 provides versatile packaging techniques beyond the capabilities of monolithic single–media systems. This allows rearranging and extending functionality to satisfy requirements specific to particular application areas. ISO/IEC 14478 is developed incrementally with parts 1 through 4 initially available. Other components are expected to be standardized by ISO/IEC JTC1 SC24 or other subcommittees.

ISO/IEC 14478 provides a framework within which application–defined ways of interacting with the environment can be integrated. Methods for the definition, presentation, and manipulation of both input and output objects are described. Application–supplied structuring of objects is also allowed and can, for example, be used as a basis for the development of toolkits for the creation of, presentation of, and interaction with multimedia and hyper–media documents and product model data.

ISO/IEC 14478 is able to support construction, presentation, and interaction with multiple simultaneous inputs and outputs using multiple media. Several such activities may occur simultaneously, and the application program can adapt its behaviour to make best use of the capabilities of its environment.

ISO/IEC 14478 includes interfaces for external storage, retrieval and interchange of multimedia objects.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 14478. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 14478 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 14478-2:1998, *Information technology — Computer graphics and image processing — Presentation Environment for Multimedia Objects (PREMO) — Part 2: Foundation Component.*

ISO/IEC 14478-3:1998, *Information technology — Computer graphics and image processing — Presentation Environment for Multimedia Objects (PREMO) — Part 3: Multimedia Systems Services.*

ISO/IEC 14478-4:1998, *Information technology — Computer graphics and image processing — Presentation Environment for Multimedia Objects (PREMO) — Part 4: Modelling, Rendering, and Interaction Component.*

3 Definitions

For the purposes of ISO/IEC 14478, the following definitions apply.

3.2.1 multimedia: The creation, editing, composing, and/or presentation of products consisting of any combination of *media.*

3.2.1.1 multimedia object: An *object* consisting of one or more types of media that can be presented to a user.

3.2.2 medium (plural media): A means by which information is perceived, expressed, stored, or transmitted. Examples include audio, video, (animated) graphics, images, text.

3.2.3 dynamic interactive graphics: Graphics applications where the graphics varies in real–time in response to user inputs.

3.2.4 animation: Series of pictures in a time–ordered sequence to display as a *video* medium. This covers all changes that have a visual effect. It thus includes the time–varying position, shape, colour, transparency, structure, and texture of an object, and changes in lighting, camera position, orientation, and focus, and even changes of rendering techniques.

3.2.5 text: A medium encompassing a character–based encoding only.

3.2.6 audio: A medium encompassing all forms of information transmitted by sound.

3.2.7 video: A medium encompassing a continuous series of pictures typically depicting motion or time sequenced events.

3.2.8 co–representation: A representation of information where the same information is presented in several different ways from which the most appropriate is chosen.

3.2.9 component: A PREMO component is a set of related object types and non–object types. The object and non–object types of a component are clustered into (component) profiles.

3.2.9.1 standard component: A component that is defined in one of the parts of ISO/IEC 14478, or a component that has been registered by an approved registration authority, and conforming to the rules defined for components in PREMO.

3.2.9.2 non–standard component: A component that is not a standard component, but which obeys the rules defined by PREMO for conforming components.

3.2.10 profile: A profile is set of related object types and non–object types from which objects can be instantiated, and a configuration specification which defines dependencies between object types and other profiles. Profiles offer a set of services embodied by the operations defined on its constituent object types.

3.2.10.1 basic profile (of a component): A mandatory set of object and non–object types for a specific component which shall be provided by all complying implementation of a component.

3.2.11 object: An entity that encapsulates some private *state* information or data, a set of associated operations that manipulate the data, and possibly a thread of control so that collectively they can be treated as a single unit.

3.2.11.1 object type: It defines the *operations* of objects; these operations collectively characterize the object's behaviour.

3.2.11.2 object reference: An object name which reliably denotes a particular object instance. This is a non–object.

3.2.12 state: Information encapsulated within an object that has to be remembered when one operation alters the future behaviour of future operations.

3.2.13 identity (of objects): Objects have a unique, immutable identity which provides a means to denote or refer to the object independent of its state or behaviour.

3.2.14 attribute (of objects): A pair of accessor and mutator functions, to retrieve the value of to set the value of the attribute.

3.2.14.1 retrieve only attribute: An object attribute where the mutator function to set the value, though conceptually present, does not change the value of the attribute, and raises an exception.

3.2.15 signature (of operations): An operation's signature consists of a list of parameter types, and a list of result types.

3.2.16 interface (of objects): The set of operation signatures defined for a type.

3.2.17 non–object types: Things that are not objects, e.g., integers, real numbers.

3.2.18 operation: Describes an action that can be applied to an object, using parameters.

3.2.18.1 operation invocation: Activation of an object's action, by describing its parameters, possibly causing results to be returned.

3.2.18.2 operation request: Synonym for operation invocation.

3.2.18.3 operation dispatching: The selection process which selects a specific operation implementation for execution.

3.2.18.4 operation overloading: The implementation of an operation defined for a *derived interface* supersedes the implementation of the same operation defined for a *base interface*. This effect is called operation overloading.

3.2.19 client: (of another object) An object issuing an operation request.

3.2.20 exception: Information returned if an error condition has occurred during the execution of a request of an operation.

3.2.21 controlling parameter: Special parameter conceptually present for all operations, used to control the way operations are dispatched. It refers to the object type on which the operation is defined.

3.2.22 subtyping: A relationship between types. It defines the rules by which objects of one type are determined to be acceptable in contexts expecting another type.

3.2.23 subtype: A type *S* is a subtype of another type *T* if any object of type *S* can be used in any context that expects an object of type *T*.

3.2.23.1 immediate subtype: A type *S* is an immediate subtype of another type *T*, if *T* is the immediate supertype of *S*.

3.2.24 supertype: A type *T* is a supertype of *S*, if *S* is a subtype of *T*.

3.2.24.1 multiple supertype: An object type can have more than one supertype; they are referred to as multiple supertypes.

3.2.24.2 immediate supertype: A type *T* is an immediate supertype of type *S*, if it is a supertype of *S*, and there is no type *Q* such that *Q* is a supertype of *S* and *T* is a supertype of *Q*.

3.2.25 direct instance: An object is a direct instance of a type *T*, if it is an instance of *T* and not an instance of any subtype of *T*.

3.2.26 immediate type: The type of the direct instance of an object instance.

3.2.27 type graph (of a type): The set of all supertypes of a type (including the recursively defined supertypes) plus the type itself.

3.2.28 inheritance : A notational mechanism for defining operation reuse. It is a relationship on interfaces.

3.2.28.1 multiple inheritance: A notational mechanism for defining operation reuse on multiple base interfaces.

3.2.28.2 single inheritance: As opposed to multiple inheritance; denotes an interface having only one base interface.

3.2.29 derived interface: If the interface *P* inherits from *Q*, *P* may also be referred to as a derived interface.

3.2.30 base interface: If the interface *P* inherits from *Q*, *Q* is a base interface (of *P*).

3.2.31 abstract object type: Non–instantiable object type.

3.2.32 operation request semantics: A finer control an object has to service an operation request.

3.2.32.1 operation receptor: A holder conceptually assigned to each operation in which operation requests are placed.

3.2.32.2 operation receptor mode: An immutable characteristics of an operation receptor.

3.32.2.1 synchronous operation receptor mode: One of the three possible modes of an operation receptor; callers are suspended on operation requests.

3.32.2.2 asynchronous operation receptor mode: One of the three possible modes of an operation receptor; callers are not suspended on operation requests, and the request's arguments are stored.

3.32.2.3 sampled operation receptor mode: One of the three possible modes of an operation receptor; callers are not suspended on operation requests, and only one requests argument is stored.

3.2.33 protected operation: An operation which can be invoked from within the object instance only; other object instances cannot request this operation.

3.2.34 life cycle (of objects and references): The mechanisms whereby objects and object references are created and destroyed.

3.2.34.1 life cycle facilities: The boundary between PREMO and its implementation environment, providing life cycle related services.

3.2.34.2 create facility: Facility to create objects possibly using initialization variables.

3.2.34.3 copy facility: Facility to create objects as copies of already existing objects.

3.34.3.1 shallow copy: Version of the copy facility when attribute values are set in the newly created object using the values of the attributes in the original object.

3.34.3.2 deep copy: Version of the copy facility when attribute values are set in the newly created object using the values of the attributes in the original object except for object references; in the case of object references, the referred objects are (deep) copied, and the new reference values are used to set the attributes.

3.2.34.4 cast facility: Facility to create an object reference to an already existing object, referring to a different immediate type.

3.2.34.5 destroyReference facility: Facility to destruct an object reference.

3.2.34.6 destroyObject facility: Facility to destruct an object instance.

3.2.35 type schema: A notational convention used to describe object types (see clause A.3).

3.2.36 operation schema: A notational convention used to describe an operation within a type schema.

3.2.37 component schema: A notational convention to describe components (see clause A.7).

3.2.38 profile schema: A notational convention to describe profiles within a component schema (see clause A.7).

3.2.39 generic type (schema): A notational convention used to describe a family of PREMO types, based on the general notational conventions of type schema (see clause B).

3.2.40 formal types: Symbols used in generic type schemas to denote non–specified object or non–object types.

3.2.41 actualization: A notational convention whereby generic type schema are used to define PREMO object types through replacing formal types by PREMO object or non–object types.

The following alphabetical list gives the subclause of each definition.

abstract object type	3.2.31
actualization	3.2.41
animation	3.2.4
asynchronous operation receptor mode	3.32.2.2

4 Symbols and abbreviations

CAD/CAM:	Computer-Aided Design / Computer-Aided Manufacturing.
IEC:	International Electrotechnical Commission.
IS:	International Standard.
ISO:	International Organization for Standardization.
JTC:	Joint Technical Committee.
PREMO:	Presentation Environments for Multimedia Objects.
2.5D:	Two and a half dimensional.
2D:	Two-dimensional.
3D:	Three-dimensional.

5 Conformance

A component is defined in PREMO to be a set of related object types and non–object types and a set of profile specifications. A component is considered to offer a set of services, embodied by the operations on the objects, and may also depend on services provided by other components.

PREMO defines conformance with respect to components as follows:

a) PREMO specifies conformance rules that shall apply for any definition of a conforming component;

b) PREMO specifies conformance rules that shall apply for any implementation of a conforming component;

c) PREMO specifies conformance rules that shall apply for any implementation of a conforming PREMO system.

A conforming component shall be defined according to the rules described in clause 9. It may also include additional requirements that shall apply to implementations of the component.

An implementation of a conforming component shall provide the mandatory set of functionality designated as a basic profile for that component, realize the configuration specification defined for that component, and in addition provide for any other implementation requirements that are given as part of the component's definition.

A standard component is a conforming component defined in one of the parts of PREMO or one that has been registered by an approved registration authority.

An implementation of a conforming PREMO system (i.e., a system implemented using PREMO components) shall obey the conformance requirements of each of the components from which it is constituted.

6 Requirements for PREMO

6.1 Introduction

Technology has evolved to the point that digital media has become an inherent part of most applications. In addition, many applications use multiple presentation media simultaneously. This combination has resulted in a large number of diverse requirements. ISO/IEC 14478 is intended to address the presentation requirements of such diverse application areas as:

a) medical systems,

b) education/training,

c) virtual reality,

d) geographic information systems,

e) digital publications,

f) scientific visualization and data exploration,

g) entertainment,

h) realtime command control systems, and

i) simulation;

and such presentation and interaction techniques as:

j) animation,

k) simultaneous use of multiple media,

l) multimodal user interfaces,

m) realistic rendering (including various dimensionalities, such 2D, 2.5D, 3D, and incorporating various media, such as video, sound, and other non–visual data).

ISO/IEC 14478 provides a common underlying functional nucleus to support these application areas and presentation techniques, as well as future areas and techniques. PREMO also enables the use of interaction techniques appropriate for specific applications, such as those listed above.

PREMO provides a generic framework, into which various organizations or applications may place their specialised objects with specific behaviour, thereby enabling interoperability. In this sense, PREMO is intended to serve as middleware, coordinating multimedia components.