INTERNATIONAL STANDARD

ISO/IEC 11072

First edition
1992-10-01

Information technology – Computer graphics – Computer Graphics Reference Model

Technologies de l'information – Infographie – Modèle de référence

Reference number
ISO/IEC 11072:1992 (E)

Contents

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

Printed in Switzerland

Annexes

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 11072 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology.*

Annexes A to D of this International Standard are for information only.

Introduction

The Computer Graphics Reference Model (CGRM) describes the conceptual framework for computer graphics. Computer graphics is the creation of, manipulation of, analysis of, and interaction with pictorial representations of objects and data using computers.

The main purpose of the CGRM is to define concepts that shall be used to develop computer graphics standards. Additional purposes are to explain relations between SC24 standards and to provide a forum whereby areas outside computer graphics can identify their relationships to computer graphics.

International Standards related to computer graphics include the following existing and emerging areas:

a) Open Systems Interconnection - Basic Reference Model;
b) Virtual Terminal Protocols and Terminal Management;
c) File Transfer, Access and Management Protocols;
d) Office Document Architecture and Interchange;
e) Text and Office Systems;
f) Exchange of Product Model Data;
g) Character Sets and Coding;
h) Open Distributed Processing;
i) Image Processing and Interchange.

This International Standard shall be the basis for the development of specific standards for computer graphics and will ensure their long term coherence based on objective rational foundations. Existing computer graphics standards will not necessarily fit precisely into the Reference Model. However, experience with current standards has significantly influenced the model.

This page intentionally left blank

Information technology – Computer graphics – Computer Graphics Reference Model

1 Scope

This International Standard, the Computer Graphics Reference Model (CGRM), defines a structure within which current and future International Standards for computer graphics shall be compared and their relationships described.

This International Standard defines a set of concepts and their inter-relationships which should be applicable to the complete range of future computer graphics standards.

This International Standard may be applied to:

a) verify and refine requirements for computer graphics;
b) identify needs for computer graphics standards and external interfaces;
c) develop models based on requirements for computer graphics;
d) define the architecture of new computer graphics standards;
e) compare computer graphics standards.

This International Standard does not define how computer graphics standards shall be defined and developed. It does not specify the functional descriptions of computer graphics standards, the bindings of those standards to programming languages, or the encoding of graphical information in any coding technique or interchange format. It is neither an implementation specification for systems incorporating computer graphics, nor a basis for appraising the conformance of implementations.

2 Definitions

For the purposes of this International Standard, the following definitions apply. An alphabetical list is given at the end of this clause.

2.1 computer graphics: The creation of, manipulation of, analysis of and interaction with pictorial representations of objects and data using computers.

2.2 application: The external object that uses computer graphics. Applications are not modelled in the CGRM, but their interactions with computer graphics are modelled.

2.3 operator: The external object that observes the contents of the display and generates physical input values. Operators are not modelled in the CGRM, but their interactions with computer graphics systems are modelled.

2.4 environment: A subdivision of the CGRM at a given level of abstraction. The definition of the environment includes the definition of its data elements and processing elements. Specific names are given to the five environments: construction, virtual, viewing, logical and realization (see 3.6.1).

2.4.1 construction environment: The environment that interfaces to the application.

2.4.2 virtual environment: The environment between the construction and viewing environments.

2.4.3 viewing environment: The environment between the virtual and logical environments.

2.4.4 logical environment: The environment between the viewing and the realization environments.

2.4.5 realization environment: The environment that interfaces to the operator.

2.4.6 higher environment: An environment closer to the application.

2.4.7 lower environment: An environment closer to the operator.

2.4.8 entity: An item of information stored within an environment or passed between environments. Entities are divided into three classes: input, output and control.

2.4.9 fan-in: The merging of entities from multiple, independent sources to produce a single stream (without changing individual entities) to be processed by a single environment.

2.4.10 fan-out: The generation of multiple, independent entities from a single entity without change. The generated entities are sent to independent environments.

2.5 external interfaces: The interfaces between the computer graphics system and the outside world, the interfaces communicate with the operator, application, data capture metafile and audit trail metafile.

2.5.1 operator interface: The interface between the realization environment and the operator. This is the only interface between the operator and the graphics system.

2.5.2 application interface: The interface provided by the construction environment to the application. This is the only interface between the application and the graphics system.

2.5.3 data capture metafile: An external object for representing all or part of a data element for storage, retrieval and transmission.

2.5.3.1 export: The process of generating a data capture metafile.

2.5.3.2 import: The action of setting part or all of a data element from a data capture metafile.

2.5.4 audit trail metafile: An external object for representing the sequential flow of information across the application interface.

2.6 processing element: A process in an environment: absorption, manipulation, distribution, assembly, and emanation.

2.6.1 absorption: A process which receives entities from the next higher environment and processes them for use within its own environment. Specific names are given to absorption at each environment level: preparation, production, projection, completion and presentation.

2.6.1.1 preparation: The name given to absorption in the construction environment.

2.6.1.2 production: The name given to absorption in the virtual environment.

2.6.1.3 projection: The name given to absorption in the viewing environment.

2.6.1.4 completion: The name given to absorption in the logical environment.

2.6.1.5 presentation: The name given to absorption in the realization environment.

2.6.2 emanation: A process which emanates token store and input control entities to the next higher environment after processing them. Specific names are given to emanation at each environment level: accumulation, abstraction, elevation, generation and utilization.

2.6.2.1 accumulation: The name given to emanation in the realization environment.

2.6.2.2 abstraction: The name given to emanation in the logical environment.

2.6.2.3 elevation: The name given to emanation in the viewing environment.

2.6.2.4 generation: The name given to emanation in the virtual environment.

2.6.2.5 utilization: The name given to emanation in the construction environment.

2.6.3 distribution: A process which distributes the composition and output control entities to the next lower environment.

2.6.4 assembly: A process which receives entities from the next lower environment for use within its own environment.

2.6.5 manipulation: A process which accesses and changes the contents of data elements.

2.7 data element: A store in an environment: composition, collection store, token store, aggregation store, and environment state.

2.7.1 composition: A spatially structured set of output primitives in a given environment. Specific names are given to the composition at each environment level: model, scene, picture, graphical image and display.

2.7.1.1 model: The name given to the composition in the construction environment.

2.7.1.2 scene: The name given to the composition in the virtual environment.

2.7.1.3 picture: The name given to the composition in the viewing environment.

2.7.1.4 graphical image: The name given to the composition in the logical environment.

2.7.1.5 display: The name given to the composition in the realization environment.

2.7.2 collection store: A storage facility for collections.

2.7.2.1 collection: A set of output entities which are named and may be structured. A collection may be manipulated to produce all or part of a composition in the same environment.

2.7.3 aggregation store: A storage facility for aggregations.

2.7.3.1 aggregation: A set of input entities which are named and may be structured. An aggregation may be manipulated to produce one or more input tokens in the token store in the same environment.

2.7.4 token store: A structured set of input tokens in a given environment. Specific names are given to the token store at each environment: lexeme store, information store, selection store, directive store and instruction store.

2.7.4.1 lexeme store: The name given to the token store in the realization environment.

2.7.4.2 information store: The name given to the token store in the logical environment.

2.7.4.3 selection store: The name given to the token store in the viewing environment.

2.7.4.4 directive store: The name given to the token store in the virtual environment.

2.7.4.5 instruction store: The name given to the token store in the construction environment.

2.7.5 environment state: Entities in the environment separate from other data elements: composition, collection store, token store, aggregation store.

2.7.6 editing: The change of entities within data elements in an environment.

2.8 output primitive: An atomic unit for graphical output in a given environment. There may be more than one class of output primitive. Geometric and other properties may be bound to an output primitive at its creation or later.

2.9 input token: An atomic unit for graphical input in a given environment. There may be more than one class of input token. Geometry and other properties may be bound to an input token at its creation or later.

2.9.1 property: A value that may be used by an output primitive or input token to specify its geometry or other characteristics.

2.9.2 geometric property: A property which is subject to modification by geometric transformations.

2.9.2.1 geometric transformation: A transformation that modifies the geometry of an input token or output primitive.

2.9.2.2 geometry: A property of an input token or output primitive used to define its shape, position, orientation and extent.

2.9.3 binding: The action of assigning a property to either an output primitive or an input token.

2.9.4 unbinding: The action of un-assigning a property from either an output primitive or an input token.

2.9.5 clipping: The action of constraining the geometric shape and extent of either an output primitive or input token to be within a specified region.

The following alphabetical list gives the sub-clause of each CGRM definition.

3 The Computer Graphics Reference Model

3.1 Environment model

The CGRM defines computer graphics in terms of five abstract levels called environments: construction, virtual, viewing, logical and realization (figure 1). The internal model of each environment is identical. The symmetry between input and output in the diagram reflects a symmetry of purpose rather than a symmetry of complexity. The CGRM defines operations on data elements in each environment.

The CGRM defines computer graphics output in terms of output primitives which make up a composition that is presented to the operator. The CGRM defines computer graphics input in terms of input tokens which make up a token store that is accumulated for the application in an appropriate form. Any connection between received input and generated output is conceptually handled by the application. The application may delegate this responsibility to specific environments. To allow complex graphical compositions, the CGRM defines a storage facility—the collection store—from which compositions may be derived. Similarly, a storage facility—the aggregation store—is defined from which entries in the token store may be derived.

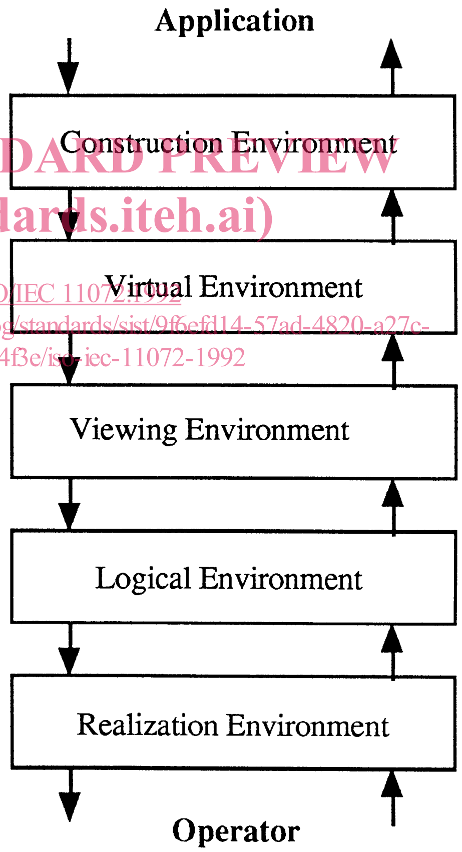

Figure 1 – Computer graphics environments