TC 97

International Standard ISO 7942

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

Information processing systems — Computer graphics — Graphical Kernel System (GKS) functional description

Systèmes de traitement de l'information — Langage de programmation graphique — GKS (Graphical Kernel System) description fonctionnelle

First edition — 1985-08-15

UDC 681.3.06 : 003.6 **Ref. No. ISO 7942-1985 (E)**

Descriptors : data processing, information interchange, graphic data processing, programming (computers), programming languages, computer graphics.

Price based on 245 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. Every 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 7942 was prepared by Technical Committee ISO/TC 97, *Information processing systems.*

Printed in Switzerland

Contents

Annexes

Information processing systems — Computer graphics — Graphical Kernel System (GKS) functional description

0 Introduction

The Graphical Kernel System (GKS) provides a set of functions for computer graphics programming. GKS is a basic graphics system that can be used by the majority of applications that produce computer generated pictures.

The main reasons for standardizing basic computer graphics are:

a) to allow application programs involving graphics to be easily portable between different installations;

b) to aid the understanding and use of graphics methods by application programmers;

c) to serve manufacturers of graphics equipment as a guideline in providing useful combinations of graphics capabilities in a device.

In order to reach these main objectives, the GKS design was based on the following requirements:

d) GKS should include all the capabilities that are essential for the whole spectrum of graphics, from simple passive output to highly interactive applications.

e) The whole range of graphics devices, including vector and raster devices, microfilm recorders, storage tube displays, refresh displays and colour displays should be controllable by GKS in a uniform way.

f) GKS should provide all the capabilities required by a majority of applications without becoming unduly large.

These requirements were used to formulate a number of principles that were used to judge specific design alternatives. Thus it was possible to contribute to the overall design goals while focussing on certain aspects. Five design aspects were identified, each having a group of principles

g) Design goals: The following principles should not be violated by any technical design:

1) consistency: the mandatory requirements of GKS should not be mutually contradictory;

2) compatibility: other standards or commonly accepted rules of practice should not be violated;

3) orthogonality: the functions or modules of GKS should be independent of each other, or the dependency should be structured and well defined.

h) Functional capabilities: The following principles were used to define the extent of GKS:

1) completeness: all functions that a majority of applications want to use on a given level of functionality should be included;

2) minimality: functions that are unnecessary for applications of a given level of functionality should not be provided;

3) compactness: an application should be able to achieve a desired result by a set of functions and parameters that is as small as possible;

4) richness: a rich set of functions offers an extensive range of facilities that stretches beyond the basic functions and includes higher order capabilities.

It is obvious that there is a trade off between the principles in this group. Therefore, the functions of GKS are organized in nine levels. An implementation of GKS provides precisely the functions of one of these levels. While the lowest level contains only a minimal set of functions, higher levels are allowed to extend beyond the basic needs towards greater richness.

i) User interface design: The following principles were used to define the user interface design:

1) user friendliness: GKS should allow the design of a desirable user interface;

2) clarity: the concepts and functional capabilities of GKS should be easily understandable, especially by the application programmer;

3) error handling: failure of system functions or modules, caused by errors of the system itself or by the application program, should be treated in such a way that the error reaction is clearly understandable and informative to the application programmer and that the impact on the system and the application program is as small as possible.

Clarity and sound error handling are essential parts of user friendliness. Error handling is an integral part of GKS. To aid clarity, the system and its state can be presented to the user in an easily comprehensible manner.

Clarity applies not only to the system design but also to the system description. To this end, the GKS specification is divided into a general description, a description of the underlying logical data structures representing the state of the system, and a description of the functions and their effects on these data structures.

j) Graphics devices: The following principles are associated with the range of graphics devices that can be addressed by GKS:

1) device independence: GKS functions should be designed to allow an application program, using these functions, to address facilities of quite different graphics output and input devices without modification of the program structure;

2) device richness: the full capabilities of a wide range of different graphics output and input devices should be accessible from the functions of GKS.

These principles led to a fundamental concept underlying the GKS architecture: the concept of multiple independent graphical workstations connected to and driven by GKS. The application program can inquire the capabilities of every workstation. The GKS design includes escape functions that are easily identifiable within an application program and can be used to access special facilities of a particular device.

k) Implementation: The last group of principles is related to the implementation of GKS:

1) implementability: it should be possible to support the GKS functions in most host languages, on most operating systems and with most graphics devices;

2) language independence: it should be possible to access the standard facilities of GKS from all ISO standard programming languages;

3) efficiency: GKS should be capable of being implemented without time consuming algorithms;

4) robustness: the operator and application programmer should be protected in the best possible way from hardware or software failure of the system.

The five groups of principles are interconnected. For example, design goals and functional capabilities both contribute to user friendliness. Efficiency is also important when considering response time in an interactive environment. Some principles may be conflicting, such as richness versus minimality, comprehensive error handling versus efficiency, and compactness versus device richness. Compromises needed to be made to achieve the overall design objective: GKS should have an easily comprehensible structure and a set of functions that enables a vast majority of computer graphics users to design portable, device independent application programs addressing the whole range of computer graphics equipment.

1 Scope and field of application

This International Standard specifies a set of functions for computer graphics programming, the Graphical Kernel System (GKS). GKS is a basic graphics system for applications that produce computer generated two dimensional pictures on line graphics or raster graphics output devices. It supports operator input and interaction by supplying basic functions for graphical input and picture segmentation. It allows storage and dynamic modification of pictures. A fundamental concept in GKS is the workstation, consisting of a number of input devices and a single output device. Several workstations can be used simultaneously. The application program is allowed to adapt its behaviour at a workstation to make best use of workstation capabilities. This International Standard includes functions for storage on and retrieval from an external graphics file. Last but not least, the functions are organized in upward compatible levels with increasing capabilities.

NOTE - For certain parameters of the functions, GKS defines value ranges as being reserved for registration (see 4.1.2). The meanings of these values will be defined using the established procedures.

GKS defines a language independent nucleus of a graphics system. For integration into a programming language, GKS is embedded in a language dependent layer obeying the particular conventions of that language.

Annexes C to G are given for information; they do not form part of the specification.

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 2382/13, *Data processing - Vocabulary - Part 13: Computer graphics.*

ISO 6093, *Information processing - Representation of numerical values in character strings for information interchange.*[1]

ISO 8632, *Information processing systems - Computer Graphics - Metafile for transfer and storage of picture description information*

- *Part 1 : Functional description.*[1]
- *Part 2 : Character encoding.*[1]
- *Part 3 : Binary encoding.*[1]
- *Part 4 : Clear text encoding.*[1]

[1] At present at the stage of draft.

3 Definitions

This clause gives the definition of the important terms of the Graphical Kernel System (GKS).

NOTE - As far as possible, commonly accepted graphics terminology is used.

3.1 acknowledgement: Output to the operator of a logical input device indicating that a trigger has fired.

3.2 aspect ratio: A ratio of x to y used to describe the shape of a rectangle in a particular coordinate system (for example, of a workstation window or a workstation viewport).

3.3 aspects of primitives: Ways in which the appearance of a primitive can vary. Some aspects are controlled directly by primitive attributes, some are controlled indirectly through a bundle table. Primitives inside segments have an aspect controlled through the segment containing them, for example highlighting; primitives outside segments do not.

3.4 attribute: A particular property that applies to a display element (output primitive) or a segment. Examples: highlighting, character height. In GKS, some properties of workstations are called workstation attributes.

3.5 baseline: A horizontal line within a character body (see figure 3) which, for many character definitions, has the appearance of being a lower limit of the character shape. A descender passes below this line. All baselines in a font are in the same position in the character bodies.

3.6 bundle index: An index into a bundle table for a particular output primitive. It defines the workstation dependent aspects of the primitive.

3.7 bundle table: A workstation dependent table associated with a particular output primitive. Entries in the table specify all the workstation dependent aspects of a primitive. In GKS, bundle tables exist for the following output primitives: polyline, polymarker, text and fill area.

3.8 capline: A horizontal line within a character body (see figure 3) which, for many character definitions, has the appearance of being the upper limit of the character shape. An ascender may pass above this line and in some languages an additional mark (for example an accent) over the character may be defined above this line. All caplines in a font are in the same position in the character bodies.

3.9 cell array: A GKS output primitive consisting of a rectangular grid of equal size rectangular cells, each having a single colour.

NOTE - These cells do not necessarily map one-to-one with pixels.

3.10 centreline: A vertical line bisecting the character body (see figure 3).

3.11 character body: A rectangle used by a font designer to define a character shape (see figure 3). All character bodies in a font have the same height.

3.12 choice device: A GKS logical input device providing a non-negative integer defining one of a set of alternatives.

3.13 clipping: Removing parts of display elements that lie outside a given boundary, usually a window or viewport.

3.14 colour table: A workstation dependent table, in which the entries specify the values of the red, green and blue intensities defining a particular colour.

3.15 coordinate graphics; line graphics: Computer graphics in which display images are generated from display commands and coordinate data.

3.16 device coordinate (DC): A coordinate expressed in a coordinate system that is device dependent. In GKS, DC units are metres on a device capable of producing a precisely scaled image and appropriate workstation dependent units otherwise.

3.17 device driver: The device dependent part of a GKS implementation intended to support a graphics device. The device driver generates device dependent output and handles device dependent interaction.

3.18 device space: The space defined by the addressable points of a display device.

3.19 display device; graphics device: A device (for example refresh display, storage tube display, plotter) on which display images can be represented.

3.20 display image; picture: A collection of output primitives or segments that are represented together at any one time on a display surface.

3.21 display space: (1) That portion of the device space corresponding to the area available for displaying images. (2) The working space of an input device such as a digitiser.

3.22 display surface; view surface: In a display device, that medium on which display images may appear.

3.23 echo: The immediate notification of the current value provided by an input device to the operator at the display console.

3.24 escape: A function in GKS used to access implementation or device dependent features, other than for the generation of graphical output, that are not otherwise addressed by GKS.

3.25 feedback: Output indicating to the operator the application program's interpretation of a logical input value.

3.26 fill area: A GKS output primitive consisting of a polygon (closed boundary) which may be hollow or may be filled with a uniform colour, a pattern, or a hatch style.

3.27 fill area bundle table: A table associating specific values for all workstation dependent aspects of a fill area primitive with a fill area bundle index. In GKS, this table contains entries consisting of interior style, style index, and colour index.

3.28 Generalized Drawing Primitive (GDP): An output primitive used to address special geometrical workstation capabilities such as curve drawing.

3.29 GKS level: Two values in the range 0 to 2 and a to c which together define a set of functional capabilities of GKS. An implementation of GKS provides precisely the functions of one level.

3.30 GKS Metafile (GKSM): A sequential file that can be written or read by GKS and is used for long-term storage (and for transmittal and transferral) of graphical information.

3.31 halfline: A horizontal line between the capline and the baseline within the character body (see figure 3), about which a horizontal string of characters in a font would appear centrally placed in the vertical direction. All halflines in a font are in the same position in the character bodies.

3.32 hatch: One possible method of filling the interior of a polygon specified by a fill area primitive. The interior is filled with an arrangement of one or more sets of parallel lines.

3.33 highlighting: A device independent way of emphasizing a segment by modifying its visual attributes. For example, blinking.

3.34 implementation mandatory: Implementation mandatory describes a property that is required to be realized identically on all workstations of all implementations of GKS.

3.35 input class: A set of input devices that are logically equivalent with respect to their function. In GKS, the input classes are: LOCATOR, STROKE, VALUATOR, CHOICE, PICK and STRING.

3.36 inquiry function: A GKS function whose purpose is to return values depending on the current state of GKS or on some fixed property of the GKS implementation. There is no effect on the state of GKS or on the display image.

3.37 locator device: A GKS logical input device providing a position in world coordinates and a normalization transformation number.

3.38 logical input device: A logical input device is an abstraction of one or more physical devices that delivers logical input values to the program. Logical input devices in GKS can be of type LOCATOR, STROKE, VALUATOR, CHOICE, PICK and STRING.

3.39 logical input value: A value delivered by a logical input device.

3.40 marker: A glyph with a specified appearance which is used to identify a particular location.

3.41 measure: A value (associated with a logical input device), which is determined by one or more physical input devices and a mapping from the values delivered by the physical devices. The logical input value delivered by a logical input device is the current value of the measure.

3.42 MI: An abbreviation for GKS metafile input, a category of workstation.

3.43 MO: An abbreviation for GKS metafile output, a category of workstation.

3.44 normalization transformation; viewing transformation; window-to-viewport transformation: A transformation that maps the boundary and interior of a window to the boundary and interior of a viewport. In GKS, this transformation maps positions in world coordinates to normalized device coordinates.

3.45 normalized device coordinates (NDC): A coordinate specified in a device independent intermediate coordinate system, normalized to some range, typically 0 to 1. In GKS, during an intermediate state the coordinates may lie outside the defined range, but associated clipping information ensures that the output does not exceed the coordinate range [0,1] × [0,1].

3.46 operator: Person manipulating physical input devices so as to change the measures of logical input devices and cause their triggers to fire.

3.47 output primitive; graphic primitive; display element: A basic graphic element that can be used to construct a display image. Output primitives in GKS are polyline, polymarker, text, fill area, cell array, and generalized drawing primitive.

3.48 pick device: A GKS logical input device providing the pick identifier attached to an output primitive and the associated segment name.

3.49 pick identifier: A name, attached to individual output primitives within a segment, and returned by the pick device. The same pick identifier can be assigned to different output primitives.

3.50 pixel; picture element: The smallest element of a display surface that can be independently assigned a colour or intensity.

3.51 polyline: A GKS output primitive consisting of a set of connected lines.

3.52 polyline bundle table: A table associating specific values for all workstation dependent aspects of a polyline primitive with a polyline bundle index. In GKS, this table contains entries consisting of linetype, linewidth scale factor, and colour index.

3.53 polymarker: A GKS output primitive consisting of a set of locations, each to be indicated by a marker.

3.54 polymarker bundle table: A table associating specific values for all workstation dependent aspects of a polymarker primitive with a polymarker bundle index. In GKS, this table contains entries consisting of marker type, marker size scale factor, and colour index.

3.55 primitive attribute: Primitive attribute values (for output primitives) are selected by the application in a workstation independent manner, but can have workstation dependent effects.

3.56 prompt: Output to the operator indicating that a specific logical input device is available.

3.57 raster graphics: Computer graphics in which a display image is composed of an array of pixels arranged in rows and columns.

3.58 rotation: Turning all or part of a display image about an axis. In GKS, this capability is restricted to segments.

3.59 scaling; zooming: Enlarging or reducing all or part of a display image by multiplying the coordinates of display elements by a constant value. In GKS, this capability is restricted to segments.

NOTE - For different scaling in two orthogonal directions two constant values are required.

3.60 segment: A collection of display elements that can be manipulated as a unit.

3.61 segment attributes: Attributes that apply only to segments. In GKS, segment attributes are visibility, highlighting, detectability, segment priority, and segment transformation.

3.62 segment priority: A segment attribute used to determine which of several overlapping segments takes precedence for graphic output and input.

3.63 segment transformation: A transformation that causes the display elements defined by a segment to appear with varying position (translation), size (scale), and/or orientation (rotation) on the display surface.

3.64 string device: A GKS logical input device providing a character string as its result.

3.65 stroke device: A GKS logical input device providing a sequence of points in world coordinates, and a normalization transformation number.

3.66 text: A GKS output primitive consisting of a character string.

3.67 text bundle table: A table associating specific values for all workstation dependent aspects of a text primitive with a text bundle index. In GKS, this table contains entries consisting of text font and precision, character expansion factor, character spacing and colour index.

3.68 text font and precision: An aspect of text in GKS, having two components, font and precision, which together determine the shape of the characters being output, on a particular workstation. In addition, the precision describes the fidelity with which the other text aspects match those requested by an application program. In order of increasing fidelity, the precisions are: STRING, CHAR and STROKE.

3.69 translation; shift: The application of a constant displacement to the position of all or part of a display image. In GKS, this capability is restricted to segments.

3.70 trigger: A physical input device or set of devices that an operator can use to indicate significant moments in time.

3.71 valuator device: A GKS logical input device providing a real number.

3.72 viewport: An application program specified part of normalized device coordinate space. In GKS, this definition is restricted to a rectangular region of normalized device coordinate space used in the definition of the normalization transformation.

3.73 window: A predefined part of a virtual space. In GKS, this definition is restricted to a rectangular region of the world coordinate space used for the definition of the normalization transformation.

3.74 workstation: GKS is based on the concept of abstract graphical workstations, which provide the logical interface through which the application program controls physical devices.

3.75 Workstation Dependent Segment Storage (WDSS): Segment storage on a workstation that is used for graphical output. Segments cannot be transferred from WDSS to another workstation.

3.76 Workstation Independent Segment Storage (WISS): A special workstation type, where segments can be stored and later transferred to other workstations.

3.77 workstation mandatory: Workstation mandatory describes a property that is required to be realized identically on all workstations of a GKS implementation.

3.78 workstation transformation: A transformation that maps the boundary and interior of a workstation window into the boundary and interior of a workstation viewport (part of display space), preserving aspect ratio. In GKS, this transformation maps positions in normalized device coordinates to device coordinates. The effect of preserving aspect ratio is that the interior of the workstation window may not map to the whole of the workstation viewport.

3.79 workstation viewport: A portion of display space currently selected for output of graphics.

3.80 workstation window: A rectangular region within the normalized device coordinate system which is represented on a display space.

3.81 world coordinate (WC): A device independent Cartesian coordinate system used by the application program for specifying graphical input and output.

4 The Graphical Kernel System

4.1 About this standard

4.1.1 Specification

The set of functions known as the Graphical Kernel System shall be as described in clauses 4, 5 and 6 and annexes A and B. These functions are organized in nine upward compatible levels with increasing capabilities as described in 4.10. An implementation of GKS shall implement precisely the functions of one level. A GKS implementation shall be invalid if it lies between two defined levels or outside the defined levels. In an implementation, all graphical capabilities that can be addressed by GKS functions shall be used only via GKS.

4.1.2 Registration[1]

For certain parameters of the functions, GKS defines value ranges as being reserved for registration. The meanings of these values will be defined using the established procedures. These procedures do not apply to values and value ranges defined as being workstation or implementation dependent; these values and ranges are not standardized.

[1] Information concerning the Registration Authority and its procedures may be obtained on request to the Secretary General, ISO Central Secretariat, case postale 56, CH-1211 Genève, Switzerland, quoting the number of this International Standard.