INTERNATIONAL STANDARD

ISO/IEC 7942-1

Second edition
1994-11-01

Information technology — Computer graphics and image processing — Graphical Kernel System (GKS) —

Part 1:
Functional description

Technologies de l'information — Infographie et traitement d'image — Système graphique GKS —

Partie 1: Description fonctionnelle

Reference number
ISO/IEC 7942-1:1994(E)

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

Contents

Foreword viii
Introduction ix
1 Scope 1
2 Normative references 2
3 Definitions 3
4 Conformance 9
4.1 Specification 9
4.2 Registration 9
5 Concepts 10
5.1 Pictures 10
5.2 Output primitive classes and attributes 10
5.3 Workstations 10
5.4 Coordinate systems and transformations 14
5.5 Logical input devices 14
5.6 Picture part store 14
5.7 State lists 14
5.8 Description tables 14
5.9 Metafiles and archives 14
5.10 Routing to the backdrop 15
5.11 Audit trail 15
5.12 Segments 15
5.13 Compatibility with ISO 7942:1985 15
5.14 Summary 15
6 The Graphical Kernel System 17
6.1 Initialization 17
6.2 Graphical output 17
6.2.1 Output primitive classes 17
6.2.2 Output primitive attributes 17
6.3 Normalization transformations 19
6.4 Picture part store 20
6.4.1 Picture part creation 20
6.4.2 Picture part functions 20
6.4.3 Adding picture parts 20
6.4.4 Picture part archiving 20
6.5 The NDC picture 20
6.5.1 Introduction 20
6.5.2 NDC picture operations 21
6.5.3 NDC metafiles 21
6.6 Selection criterion 21
6.7 Graphical input 22
6.7.1 Introduction to logical input devices 22
6.7.2 Logical input device model 23
6.7.3 Operating modes of logical input devices 24
6.7.4 Input queue and current event report 25

© ISO/IEC 1994
All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

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

Printed in Switzerland

Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

6.8 Inquiry functions....26
6.9 Error handling....26
6.10 Special interfaces between GKS and the application program....26
6.11 Backdrop....26
6.12 Audit and playback....27
7 Workstation dependent control....28
7.1 Introduction....28
7.2 Workstation characteristics....28
7.3 Selecting a workstation....29
7.4 Selection criteria....29
7.5 Viewing....30
7.6 Workstation transformations....30
7.7 Output primitives....31
7.8 Colour....32
7.9 Setting representations....33
7.10 Removing a backdrop....34
7.11 Visual effect state....34
7.12 Realized metafile....34
7.13 Logical input devices....34
7.13.1 Introduction....34
7.13.2 Initialization of logical input devices....35
7.13.3 Definition of logical and composite input devices....35
7.14 Sending messages to a workstation....35
8 Output primitives....36
8.1 Introduction....36
8.2 Curve output primitives....36
8.2.1 Curve primitives....36
8.2.2 Curve attributes....37
8.3 Marker output primitives....37
8.4 Area output primitives....38
8.4.1 Area primitives....38
8.4.2 Definition of interior....38
8.4.3 Area attributes....38
8.5 Character output primitives....40
8.5.1 Introduction....40
8.5.2 Glyph specification....41
8.5.3 Glyph size....42
8.5.4 Text extent....42
8.5.5 Text skewing....43
8.5.6 Text alignment....44
8.5.7 Text orientation....46
8.5.8 Transformed text....47
8.5.9 Precision....47
8.5.10 Estimate of text extent....48
8.6 Image output primitives....48
8.7 Design output primitives....49
8.7.1 Introduction....49
8.7.2 Stencils....50
8.7.3 Stencil attributes....52
8.7.4 Stencil composition....53
8.7.5 Tiling store....57
8.7.6 Stencil and tiling functions....59

AVA Documents
Document Sharing Hub
Full document details: ISO IEC 07942-1-1994

8.8 Generalized drawing primitive 59
9 Logical input device classes 60
9.1 Introduction.... 60
9.2 Measures of each logical input device class....60
9.3 Transformation of LOCATOR and STROKE input....61
9.3.1 Transformation of LOCATOR input....61
9.3.2 Transformation of STROKE input 63
9.4 Prompt and echo types....63
9.4.1 LOCATOR prompt and echo types....63
9.4.2 STROKE prompt and echo types 64
9.4.3 VALUATOR prompt and echo types....64
9.4.4 CHOICE prompt and echo types....65
9.4.5 PICK prompt and echo types....65
9.4.6 STRING prompt and echo types 66
10 Segments and workstation activation 67
10.1 Introduction.... 67
10.2 Selection criteria 67
10.3 Segment state list....68
10.4 Workstation activation....68
10.5 Segment creation 68
10.6 Segment manipulation 68
10.7 Segment attributes 69
10.8 Segment storage....69
10.9 Clear workstation....69
11 Data types 70
11.1 Data type definitions....70
11.1.1 Notational conventions 70
11.1.2 Basic types 72
11.1.3 Data types for GKS control 73
11.1.4 Data types for output primitives 73
11.1.5 Data types for output attributes 75
11.1.6 Data types for transformations 78
11.1.7 Data types for NDC picture....79
11.1.8 Data types for metafiles 79
11.1.9 Data types for picture parts and archives 79
11.1.10 Data types for utility functions 80
11.1.11 Data types for segments....80
11.1.12 Data types for input 81
11.1.13 Data types for workstation control 82
11.1.14 Data types for inquiry functions 83
11.1.15 Data types for operating state 83
11.1.16 Data types for font and glyph functions 84
11.1.17 Data types for audit trails....84
11.1.18 Data types for selection criteria....84
11.1.19 Data types for paths, tilings and stencils 85
11.2 Data type definitions for state lists and description tables....87
11.2.1 Introduction 87
11.2.2 Operating state list (OSL)....87
11.2.3 GKS description table (GDT)....87
11.2.4 GKS state list (GSL)....87
11.2.5 Workstation state list (WSL) 88
11.2.6 Workstation description table (WDT) 88

Documents
Document Sharing Hub
Full document details: ISO IEC 07942-1-1994

11.2.7 Error state list (ESL) 90
11.2.8 Segment state list (SSL) 90
11.2.9 Stencil state list (STSL) 90
11.3 Initial values of state lists and description tables 90
11.3.1 Operating state list 90
11.3.2 GKS description table 90
11.3.3 GKS state list 90
11.3.4 Workstation state list 92
11.3.5 Generic workstation description table 92
11.3.6 Error state list 92
11.3.7 Segment state list 92
11.3.8 Stencil state list 92
12 Workstation independent functions 93
12.1 Control functions 93
12.2 Output functions 94
12.3 Design output functions 95
12.4 Output attributes 97
12.5 Normalization transformation functions 98
12.6 NDC picture functions 99
12.7 Metafile functions 100
12.8 Picture part store functions 100
12.9 Input functions 103
12.10 Font and glyph functions 104
12.11 Audit and playback functions 104
12.12 Inquiry functions 105
12.13 Utility functions 107
12.14 Utility functions for output primitives 107
13 Workstation functions 109
13.1 Control functions 109
13.2 Inquiry functions 112
13.3 Retrieval functions 114
13.4 Viewing utility functions 115
13.5 Colour utility functions 115
14 Segment and workstation activation functions 116
14.1 Segment functions 116
14.2 Workstation activation functions 118
14.3 Utility functions 119
Annexes
A Function and data type list 120
A.1 Functions alphabetic 120
A.2 Functions order of appearance 122
A.3 Data types alphabetic 126
B Error list 128
B.1 Function error list 128
B.2 Error list ordered by error number 138
C Language binding considerations 140
D Allowable differences 146
D.1 General 146
D.2 Global differences 146
D.3 Workstation dependent differences 146
E Colour models 149
F Bibliography 150

AVA Documents
Document Sharing Hub
Full document details: ISO IEC 07942-1-1994

G An approach to compatibility with ISO 7942:1985 151
G.1 Introduction 151
G.2 Format 151
G.2.1 Compatibility aspects 151
G.2.2 Language binding 151
G.2.3 Notation used in this annex 151
G.3 Data structures 151
G.3.1 Introduction 151
G.3.2 Compatibility data types 152
G.3.3 Compatibility operating state list (COSL) 152
G.3.4 Compatibility GKS state list (CGSL) 152
G.3.5 Compatibility workstation state list (CWSL) 152
G.3.6 Compatibility segment state list (CSSL) 153
G.4 Control functions 153
G.4.1 OPEN GKS 153
G.4.2 OPEN WORKSTATION 153
G.4.3 CLOSE WORKSTATION 154
G.4.4 ACTIVATE WORKSTATION 154
G.4.5 DEACTIVATE WORKSTATION 155
G.4.6 CLEAR WORKSTATION 155
G.4.7 Deferral states 155
G.5 Output and attribute functions 156
G.6 Transformation functions 156
G.6.1 SET VIEWPORT 156
G.6.2 SET WINDOW 157
G.6.3 SELECT NORMALIZATION TRANSFORMATION 157
G.7 Workstation transformation functions 157
G.8 Segment functions 157
G.8.1 Introduction 157
G.8.2 CREATE SEGMENT 157
G.8.3 DELETE SEGMENT FROM WORKSTATION 157
G.8.4 ASSOCIATE SEGMENT WITH WORKSTATION 158
G.8.5 COPY SEGMENT TO WORKSTATION 158
G.8.6 INSERT SEGMENT 158
G.8.7 Segment attributes 158
G.9 Input functions 158
G.9.1 Introduction 158
G.9.2 Simultaneous events 159
G.9.3 Pick input 160
G.10 GKSM 160
G.10.1 READ ITEM FROM GKSM 160
G.10.2 GET ITEM FROM GKSM 160
G.10.3 INTERPRET ITEM 160
G.10.4 WRITE ITEM TO GKSM 161
G.11 Inquiry functions 161
G.11.1 Inquiry functions for operating state value 161
G.11.2 Inquiry functions for GKS description table 161
G.11.3 Inquiry functions for GKS state list 161
G.11.4 Inquiry functions for workstation state list 161
G.11.5 Inquiry functions for workstation description table 161

Documents
Document Sharing Hub
Full document details: ISO IEC 07942-1-1994

G.11.6 Inquiry functions for segment state list 162
G.11.7 Pixel inquiries 162
G.11.8 Inquiry functions for GKS error state list 162
G.12 Utility functions 162
G.13 Error handling 162
G.13.1 Matching error numbers 162
G.13.2 Using the utility function 162
G.14 Conclusions 163
H Compatibility with the Computer Graphics Reference Model ISO/IEC 11072: 1992 (informative) 164
H.1 Introduction 164
H.2 Environments 164
H.3 Fan-out and fan-in 164
H.4 External interfaces 164
H.5 Processing elements 164
H.6 Compositions 164
H.7 Collections 164
H.8 Clipping 164

AVA Documents
Document Sharing Hub
Full document details: ISO IEC 07942-1-1994

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 JTC1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as International Standard requires approval by at least 75% of the national bodies casting a vote.

International Standard ISO/IEC 7942-1 was prepared by Joint Technical Committee ISO/IEC JTC1, *Information technology, Sub-Committee 24, Computer graphics and image processing.*

This second edition cancels and replaces the first edition (ISO 7942:1985), which has been technically revised.

ISO/IEC 7942 consists of the following parts, under the general title *Information technology – Computer graphics and image processing – Graphical Kernel System (GKS):*

Part 1: Functional description

Part 2: NDC metafile

Part 3: Audit trail

Part 4: Picture part archive

Annexes A and B form an integral part of this part of ISO/IEC 7942. Annexes C, D, E, F, G, and H are for information only.

Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

Introduction

The Graphical Kernel System (GKS) provides a set of functions for computer graphics programming that can be used by a range of applications. The main motivations for standardization are to improve portability of programs and to define a basic methodology. Portability is achieved by providing all the capabilities necessary in a device independent way. The following principles are used in specifying GKS:

a) Design: the three goals are consistency of approach, compatibility with related standards and orthogonal functionality where possible.

b) Functionality: the goals are completeness with the minimal set of functions. Organization of functions should be such as to achieve compact programs. Richness should be provided by utilities and toolkits on top of GKS rather than non-standard extensions to GKS.

c) Clarity: the underlying concepts should be easily understood, especially by the application programmer. To achieve this, GKS is defined using a small set of functions with precise specification of the data structures that define the state of GKS and the effect that functions have on this state.

d) Error handling: all errors caused by incorrect function invocations or internal failures are logged with the application having control over the action taken.

e) Separation of device dependent functionality: in GKS there is a clear separation between the functionality that is device dependent and device independent. GKS has the concept of a workstation which deals with all device dependent functions.

f) Implementation: GKS should be realizable in a wide range of host languages. The support GKS requires from the operating system should not be excessive. GKS should be efficient to implement on commonly available hardware and it should be easy to produce a robust product.

This International Standard (known informally as "GKS-94") has a strong relationship to the principles and functionality of ISO 7942:1985 (known informally as "GKS-85"). However, it has been updated to provide additional functionality, cleaner concepts and an International Standard for modern hardware. Minor changes made include some additional primitives, some minor extensions to input, a clearer definition of the NDC picture and simplification of deferral.

A nameset attribute has been added and this is used as a selection criterion for a range of functions. As a result, the equivalent of the ISO 7942:1985 segment (picture part) no longer needs attributes separate from the primitive attributes. The segment facilities of ISO 7942:1985 are provided in terms of the new picture part store.

The ISO 7942:1985 metafile is replaced by an audit trail and an NDC metafile based on the Computer Graphics Metafile (ISO/IEC 8632).

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

Information technology – Computer graphics and image processing - Graphical Kernel System (GKS) – Part 1: Functional description

1 Scope

This part of ISO/IEC 7942 specifies a set of functions for computer graphics programming, the Graphical Kernel System (GKS). It provides functions for two dimensional graphical output, the storage and dynamic modification of pictures, and operator input. GKS functions and datatypes are specified independently of programming languages.

GKS establishes a system for device independent graphics programming by separating picture composition and interaction from the realization of the pictures on a specific output device and the input devices used by the operator.

This International Standard is applicable to a wide range of applications that produce two dimensional pictures on vector or raster graphical devices in monochrome or colour. Operator interaction is allowed with these pictures.

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 7942. 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 7942 are encouraged to investigate the possibilities of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 646:1991, *Information technology - ISO 7-bit coded character set for information interchange.*

ISO 2022:1986, *Information processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques.*

ISO 2382-13:1984, *Data processing - Vocabulary - Part 13: Computer graphics.*

ISO 6093:1985, *Information processing - Representation of numerical values in character strings for information interchange.*

ISO/IEC 8632:1992, *Information technology - Computer graphics - Metafile for transfer and storage of picture description information*

- *Part 1 : Functional specification.*
- *Part 2 : Character encoding.*
- *Part 3 : Binary encoding.*
- *Part 4 : Clear text encoding.*

ISO/IEC 9541:1991, *Information technology - Font information interchange.*

Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

3 Definitions

For the purposes of this part of ISO/IEC 7942, the following definitions apply.

3.1 picture: A spatially structured sequence of output primitives destined for storage or display on a workstation. The following pictures exist: NDC picture, logical picture, realized picture.

3.1.1 NDC picture: The picture in which graphical output is composed by the application program and with which the operator interacts using graphical input devices.

3.1.2 logical picture: The picture constructed from the NDC picture for a particular workstation in which logical attributes are bound to output primitives.

3.1.3 realized picture: The picture constructed from the logical picture for a particular workstation by binding the precise definition of colours specified either directly or indirectly. The realized picture is displayed on the workstation's display space.

3.2 workstation: A display space and associated input peripherals.

3.2.1 display surface: The physical area on a display device onto which display spaces are mapped.

3.2.2 display space: The area available for displaying images on a particular workstation.

3.3 coordinate system: The metric used to specify graphical output and positional input. The coordinate systems are world coordinates, normalized device coordinates, logical device coordinates and device coordinates. All coordinate systems are cartesian.

3.3.1 world coordinates (WC): The coordinates used by the application program to define output primitives and positional input.

3.3.2 normalized device coordinates (NDC): The coordinates used to define the NDC picture and picture parts.

3.3.3 logical device coordinates (LDC): The coordinates used to define the logical picture.

3.3.4 device coordinates (DC): The coordinates used to define realized pictures on a workstation.

3.4 transformation: A process for transforming one coordinate system to another.

3.4.1 normalization transformation: A window-to-viewport transformation that maps positions in WC to NDC.

3.4.2 normalization transformation number: An identifier of a particular normalization transformation.

3.4.3 view orientation: A transformation that transforms the NDC coordinate system to one more appropriate for the object to be viewed.

3.4.4 view mapping: A transformation applied after view orientation to map to LDC coordinates.

3.4.5 workstation transformation: A workstation window-to-viewport mapping that maps positions in LDC to device coordinates, preserving aspect ratio.

3.5 scissoring: Restricting the appearance of an output primitive on a display by either clipping or shielding.

3.5.1 view scissoring: Scissoring applied to output primitives in a view after view orientation and view mapping. View scissoring may be postponed until after the workstation transformation.

3.6 picture part store: A collection of picture parts.

3.6.1 picture part: A sequence of output primitives with associated attributes.

3.7 state list: A data structure whose entries specify the current values of variables relating to GKS as a whole (GKS state list) or to a specific workstation (workstation state list).

3.8 description table: A data structure whose entries specify the capabilities of an implementation (GKS description table) or a type of workstation (generic workstation description table) or an instance of a type of workstation (workstation description table).

AVA Documents
Document Sharing Hub
Full document details: ISO IEC 07942-1-1994

3.9 backdrop: Graphical output that appears on the workstation but is not part of the NDC picture, logical picture, or realized picture. The backdrop arrives at the display device on a separate channel from the realized picture. It is device dependent how the two appear relative to each other.

3.9.1 router: A switch which controls the destination to which output primitives are dispatched.

3.10 output primitive: A basic graphical element used to construct the NDC picture and picture parts.

3.11 audit trail: The external entity for representing the sequential flow of information across the interface between the application and GKS.

3.12 output primitive classes: A classification associated with output primitives. A set of logical attributes apply to all output primitive types in a class. The output primitive classes are curve, marker, area, character, image, design. The output primitive, generalized drawing primitive, does not belong to any class.

3.12.1 curve: A class of output primitive which generates a set of curves. The output primitives in the class are set of polyline, set of nurb and set of conic section.

3.12.1.1 set of polyline: A type of output primitive consisting of a set of polylines each of which consists of a connected sequence of straight lines between specified points.

3.12.1.2 set of nurb: A type of output primitive consisting of a set of curves each of which is defined by a nurb.

3.12.1.2.1 nurb: A curve defined by a non-uniform B-spline. Both rational and non-rational versions are allowed.

3.12.1.3 set of conic section: A type of output primitive consisting of a set of curves each of which is defined by a conic section.

3.12.2 marker: A class of output primitive which generates a set of symbols. The only output primitive in the class is polymarker.

3.12.2.1 polymarker: A type of output primitive consisting of a set of positions at which symbols are placed.

3.12.3 area: A class of output primitive which generates a set of areas with edges. The output primitives in the class are set of fill area, set of closed nurb, set of elliptic sector, set of elliptic segment and set of elliptic disc.

3.12.3.1 set of fill area: A type of output primitive consisting of a set of areas each of which is defined by a closed sequence of connected points.

3.12.3.2 set of closed nurb: A type of output primitive consisting of a set of areas each of which is defined by a closed nurb.

3.12.3.2.1 closed nurb: A nurb which has the first point connected to the end point to produce a closed curve if the two points are not coincident.

3.12.3.3 set of elliptic sector: A type of output primitive consisting of a set of areas each of which is defined by an elliptic arc. Each elliptic arc area is closed by lines from the ends of the arc to the centre of the ellipse.

3.12.3.4 set of elliptic segment: A type of output primitive consisting of a set of areas each of which is defined by an elliptic arc. Each elliptic arc area is closed by joining the ends of the arc.

3.12.3.5 set of elliptic disc: A type of output primitive consisting of a set of areas each of which is defined by a complete ellipse.

3.12.3.6 edge: The boundaries of closed regions defined in the area class of output primitives.

3.12.4 character: A class of output primitive which generates a sequence of glyphs. The only output primitive in the class is text.

3.12.4.1 text: A type of output primitive consisting of a sequence of glyphs defined relative to a position.

Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

Definitions

3.12.4.1.1 glyph: A recognizable abstract graphic symbol which is independent of any specific design (ISO/IEC 9541).

3.12.4.1.2 glyph body: A rectangle used by a font designer to define a glyph shape (see figure 1). All glyph bodies in a font have the same height.

3.12.4.1.3 glyph shape: The set of information in a glyph representation used for displaying the dimensions and positioning of the glyph shape (ISO/IEC 9541).

3.12.4.1.4 glyph representation: The glyph shape and glyph metrics associated with a specified glyph in a font resource (ISO/IEC 9541).

3.12.4.1.5 topline: A horizontal line at the top of a glyph body (see figure 1) which is just above the upper limit of all glyph shapes in a font. Ascenders and accents are below the topline.

3.12.4.1.6 capline: A horizontal line within a glyph body (see figure 1) which, for many glyph definitions, has the appearance of being the upper limit of the glyph shape. An ascender may pass above this line and in some languages an additional mark (for example an accent) over the glyph may be defined above this line. All caplines in a font are in the same position in the glyph bodies.

3.12.4.1.7 baseline: A horizontal line within a glyph body (see figure 1) which, for many glyph definitions, has the appearance of being a lower limit of the glyph shape. A descender passes below this line. All baselines in a font are in the same position in the glyph bodies.

3.12.4.1.8 halfline: A horizontal line between the capline and the baseline within the glyph body (see figure 1), about which a horizontal string of glyphs in a font would appear centrally placed in the vertical direction. All halflines in a font are in the same position in the glyph bodies.

3.12.4.1.9 bottomline: A horizontal line at the bottom of the glyph body (see figure 1) which is just below all descenders in a font. All bottomlines in a font are in the same position in the glyph bodies.

3.12.4.1.10 centreline: A vertical line bisecting the glyph body (see figure 1).

3.12.4.1.11 leftline: A vertical line at the left of a glyph body (see figure 1) which is to the left of all glyph shapes in a font.

3.12.4.1.12 rightline: A vertical line at the right of a glyph body (see figure 1) which is to the right of all glyph shapes in a font.

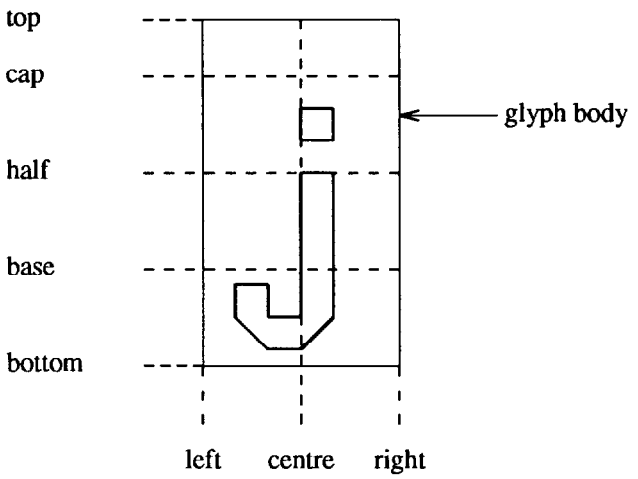

Figure 1 - Glyph coordinate system

3.12.4.1.13 font: A collection of glyph images having the same basic design (ISO/IEC 9541).

3.12.4.1.14 font resource: A collection of glyph representations together with description and font metric information which are relevant to the collection of glyph representations as a whole

AVA Documents
Document Sharing Hub
Full document details: ISO IEC 07942-1-1994

(ISO/IEC 9541).

3.12.4.1.15 posture: The extent to which the shape of a glyph or set of glyphs appear to incline, including any consequent design or form change (ISO/IEC 9541).

3.12.4.1.16 escapement Movement of the current position on the presentation surface after a glyph representation is imaged (ISO/IEC 9541).

3.12.5 image: A class of output primitive which generates images. The only output primitive in the class is cell array.

3.12.5.1 cell array: A type of output primitive defined by a rectangular grid of equal size rectangular cells, each having a single colour.

3.12.6 design: A class of output primitive which generates graphical output by extruding a tiling through a stencil. Stencils are stored in the stencil store. Tilings are stored in the tiling store.

3.12.6.1 stencil: A set of shapes. Stencils may be constructed from area boundaries (which may be defined by closed paths), closed contours and other stencils.

3.12.6.1.1 closed path: A sequence of paths where the end of one path is the start of the next and where the last point is the same as the first.

3.12.6.1.2 path: A sequence of points used to specify the locus between two end points.

3.12.6.1.3 contour: A stencil which is defined as the outline surrounding a path.

3.12.6.2 tiling: A sequence of tiling components.

3.12.6.2.1 tiling component: A replicated tile.

3.12.6.2.2 tile: A pattern defined as a curve, area or design.

3.12.7 generalized drawing primitive (GDP): A type of output primitive used to address special output requirements.

3.13 attribute: A particular property that applies to an output primitive.

3.13.1 identification attributes: A class of attribute used to name output primitives.

3.13.1.1 nameset: An identification attribute (in the form of a set of names) of an output primitive.

3.13.1.2 pick identifier: An identification attribute of an output primitive.

3.13.1.3 selection criterion: A rule for choosing elements from a sequence of output primitives. Selection criteria are expressed as operations on namesets.

3.13.2 NDC attributes: A class of attribute bound to primitives when they are created which completes the geometric definitions of primitives.

3.13.2.1 local transformation: An NDC attribute which defines a transformation in NDC space which is applied to the geometry of a primitive to position individual output primitives.

3.13.2.2 global transformation: An NDC attribute which defines a transformation in NDC space which is applied to the geometry of a primitive after the local transformation. Global transformations are used to apply a transformation to a complete object.

3.13.2.3 NDC scissor: A set of scissors applied to the NDC picture and backdrop after the local and global transformations. NDC scissoring may be postponed until after the workstation transformation.

3.13.3 source attributes: A class of attribute bound to primitives when they are created which are used to determine the values of the logical attributes which will be applied to primitives at the logical level. Bundle indices and attribute source flags are source attributes.

3.13.3.1 bundle index: An index into a particular bundle table.

3.13.3.2 attribute source flag (ASF): A flag for each logical attribute, which determines whether the value of a logical attribute of a primitive at the logical level will be taken from the logical attribute bound to the primitive at the NDC level, or from the bundle table using the bundle index bound at the NDC level. Attribute source flags for each of the logical attributes of a primitive are bound to the

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

primitive at the NDC level.

3.13.3.3 bundle table: A workstation dependent table with entries accessed by a bundle index. Each table entry defines the values of a set of logical attributes to be associated with a class of output primitive. Whether these values are bound to particular output primitives at the logical level depends on the settings of the corresponding attribute source flags.

3.13.4 logical attributes: A class of attribute which defines the appearance of an output primitive in the logical picture. Logical attributes may have different values for different workstations.

3.13.4.1 colour specifier: A colour type and colour value. Colour may be specified either directly or indirectly. If the colour type is indirect, the colour value is a colour index, otherwise the colour type defines a colour model and the colour value defines the value of the colour in that model. The realized picture defines the precise colour of primitives by substituting the appropriate definitions for indirectly specified colours.

3.13.4.2 colour array specifier: A colour type and colour value. If the colour type is indirect, the colour value is an array of colour indices. Otherwise the colour type specifies a colour model and the colour value is an array of values defining the colours in that model.

3.13.4.3 colour index: An index into a particular colour table.

3.13.4.4 colour table: A workstation dependent table. Each table entry defines, by a colour model name and associated values, the colour corresponding to a particular value of the colour index.

3.14 logical input device: An abstraction of one or more physical devices that delivers logical input values to the program. Specific logical input devices may be of class locator, stroke, valuator, choice, pick, string and composite.

3.14.1 logical input value: A measure value delivered by a logical input device.

3.14.1.1 measure: A value which is determined by one or more physical input devices and a mapping from the values delivered by the physical devices.

3.14.1.2 trigger: A condition which determines when a logical input value is delivered by a logical input device.

3.14.2 locator: A class of logical input device providing a position in world coordinates and a normalization transformation number.

3.14.3 stroke: A class of logical input device providing a sequence of positions in world coordinates and a normalization transformation number.

3.14.4 valuator: A class of logical input device providing a real number.

3.14.5 choice: A class of logical input device providing a positive integer.

3.14.6 pick: A class of logical input device providing the identification attributes of an output primitive.

3.14.7 string: A class of logical input device providing a character string.

3.14.8 composite: A class of logical input device built up from previously defined devices.

3.14.9 prompt: An indication to the operator that a specific logical input device is available.

3.14.10 echo: The immediate notification to the operator of the current value of an input device.

The following alphabetical list gives the subclause of each GKS definition.

area	3.12.3	bottomline	3.12.4.1.9
attribute	3.13	bundle index	3.13.3.1
attribute source flag (ASF)	3.13.3.2	bundle table	3.13.3.3
audit trail	3.11	capline	3.12.4.1.6
backdrop	3.9	cell array	3.12.5.1
baseline	3.12.4.1.7	centreline	3.12.4.1.10

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

character	3.12.4	normalization transformation number	3.4.2
choice	3.14.5	normalized device coordinates (NDC)	3.3.2
closed nurb	3.12.3.2.1	nurb	3.12.1.2.1
closed path	3.12.6.1.1	output primitive	3.10
colour array specifier	3.13.4.2	output primitive classes	3.12
colour index	3.13.4.3	path	3.12.6.1.2
colour specifier	3.13.4.1	pick	3.14.6
colour table	3.13.4.4	pick identifier	3.13.1.2
composite	3.14.8	picture	3.1
contour	3.12.6.1.3	picture part	3.6.1
coordinate system	3.3	picture part store	3.6
curve	3.12.1	polymarker	3.12.2.1
description table	3.8	posture	3.12.4.1.15
design	3.12.6	prompt	3.14.9
device coordinates (DC)	3.3.4	realized picture	3.1.3
display space	3.2.2	rightline	3.12.4.1.12
display surface	3.2.1	router	3.9.1
echo	3.14.10	scissoring	3.5
edge	3.12.3.6	selection criterion	3.13.1.3
escapement	3.12.4.1.16	set of closed nurb	3.12.3.2
font	3.12.4.1.13	set of conic section	3.12.1.3
font resource	3.12.4.1.14	set of elliptic disc	3.12.3.5
generalized drawing primitive (GDP)	3.12.7	set of elliptic sector	3.12.3.3
global transformation	3.13.2.2	set of elliptic segment	3.12.3.4
glyph	3.12.4.1.1	set of fill area	3.12.3.1
glyph body	3.12.4.1.2	set of nurb	3.12.1.2
glyph representation	3.12.4.1.4	set of polyline	3.12.1.1
glyph shape	3.12.4.1.3	source attributes	3.13.3
halfline	3.12.4.1.8	state list	3.7
identification attributes	3.13.1	stencil	3.12.6.1
image	3.12.5	string	3.14.7
leftline	3.12.4.1.11	stroke	3.14.3
local transformation	3.13.2.1	text	3.12.4.1
locator	3.14.2	tile	3.12.6.2.2
logical attributes	3.13.4	tiling	3.12.6.2
logical device coordinates (LDC)	3.3.3	tiling component	3.12.6.2.1
logical input device	3.14	topline	3.12.4.1.5
logical input value	3.14.1	transformation	3.4
logical picture	3.1.2	trigger	3.14.1.2
marker	3.12.2	valuator	3.14.4
measure	3.14.1.1	view mapping	3.4.4
nameset	3.13.1.1	view orientation	3.4.3
NDC attributes	3.13.2	view scissoring	3.5.1
NDC picture	3.1.1	workstation	3.2
NDC scissor	3.13.2.3	workstation transformation	3.4.5
normalization transformation	3.4.1	world coordinates (WC)	3.3.1

Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

4 Conformance

4.1 Specification

The set of functions known as GKS shall be as described in clauses 5, 6, 7, 8, 9, 10, 11, 12, 13, and 14 and annexes A and B. In an implementation all graphical capabilities of a display space that can be addressed by GKS functions shall be used only via GKS.

4.2 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 described in ISO/IEC TR 9973: 1988. These procedures do not apply to values and value ranges defined as being workstation or implementation dependent; these values and value ranges are not standardized.

1) For the purpose of this part of ISO/IEC 7942 and according to the rules for the designation and operation of registration authorities in the ISO/IEC Directives, the ISO and IEC Councils have designated the following as the registration authority: National Institute of Standards and Technology (National Computer Systems Laboratory), The Registration Authority for Graphical Items, A-266 Technology Building, Gaithersurg, MD 20899.

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

5 Concepts

5.1 Pictures

A central concept is the *picture* which is a spatially structured sequence of output primitives destined for storage or display on a workstation. The *NDC picture* (Normalized Device Coordinate picture) is where graphical output is composed and with which the operator interacts using graphical input devices (see figure 2) on one or more workstations. Alternatively, graphical output can be dispatched directly to a backdrop in which case it cannot be interacted with.

For each workstation, a number of views of the NDC picture can be selected. Each view has all the logical attributes bound to create the *logical picture* as the set of views. Finally, the precise colour definitions are bound to create the *realized picture* for display on the workstation's display space. Changes to the NDC picture will be reflected in changes to the logical and realized pictures.

The operator interacts with the set of views in the realized picture. Where these overlap, it is possible to define which view takes priority in terms of operator interactions.

5.2 Output primitive classes and attributes

Output primitive classes are abstractions of basic actions a device can perform such as drawing curves, rendering areas and writing character strings. There may be several different types of output primitive in each class. Associated with output primitives are *attributes* which define additional properties of the primitive. For example, the type of curve to be output is specified by a primitive attribute.

There are four classes of primitive attributes:

a) identification

b) NDC

c) source

d) logical

Identification attributes can be used to partition output primitives into sets for a variety of purposes. The main identification attribute is the *nameset*.

NDC attributes complete the geometric definition of the output primitive in the NDC coordinate space. All output primitives have a *local transformation*, a *global transformation* and an *NDC scissor* as NDC attributes.

Source attributes determine the source of values for the attributes of the primitives in the logical picture.

Logical attributes define the appearance of the output primitives in the logical picture to be displayed on the workstation.

When an output primitive is created, identification, NDC, logical and source attributes are bound to it.

Attributes of output primitives in the NDC picture may be changed.

5.3 Workstations

GKS defines a *workstation* as an abstract *display space* on a device's *display surface* and associated input peripherals. Multiple workstations may be in operation together. Workstations may have no associated input peripherals or no display space.

A *selection criterion* based on the namesets of the output primitives in the NDC picture and backdrop defines what subset of the NDC picture and backdrop is displayed on a workstation. A second set of selection criteria decide how many views of each output primitive selected are displayed on the workstation.

For each workstation, the NDC picture is transformed into a *logical picture* on the workstation partitioned into views. The logical picture is defined in *logical device coordinates* (LDC). Output primitives in the logical

Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

picture have all logical attributes and relevant NDC attributes bound to them.

Logical attributes are bound to output primitives in the NDC picture. The values bound to the equivalent output primitive in the logical picture depend on the source attributes. They are either the values of the logical attributes bound to the primitives in the NDC picture or the values extracted from workstation bundle tables. The attribute related to colour is the *colour specifier*.

The logical picture is transformed into the *realized picture* which is realized on the workstation display space. The realized picture is obtained by binding the required colour information, for indirectly specified colours (pointed at by the colour indices), to the output primitives in the logical picture. Figure 3 shows the NDC picture and picture part store. Figure 4 shows the logical and the realized pictures.

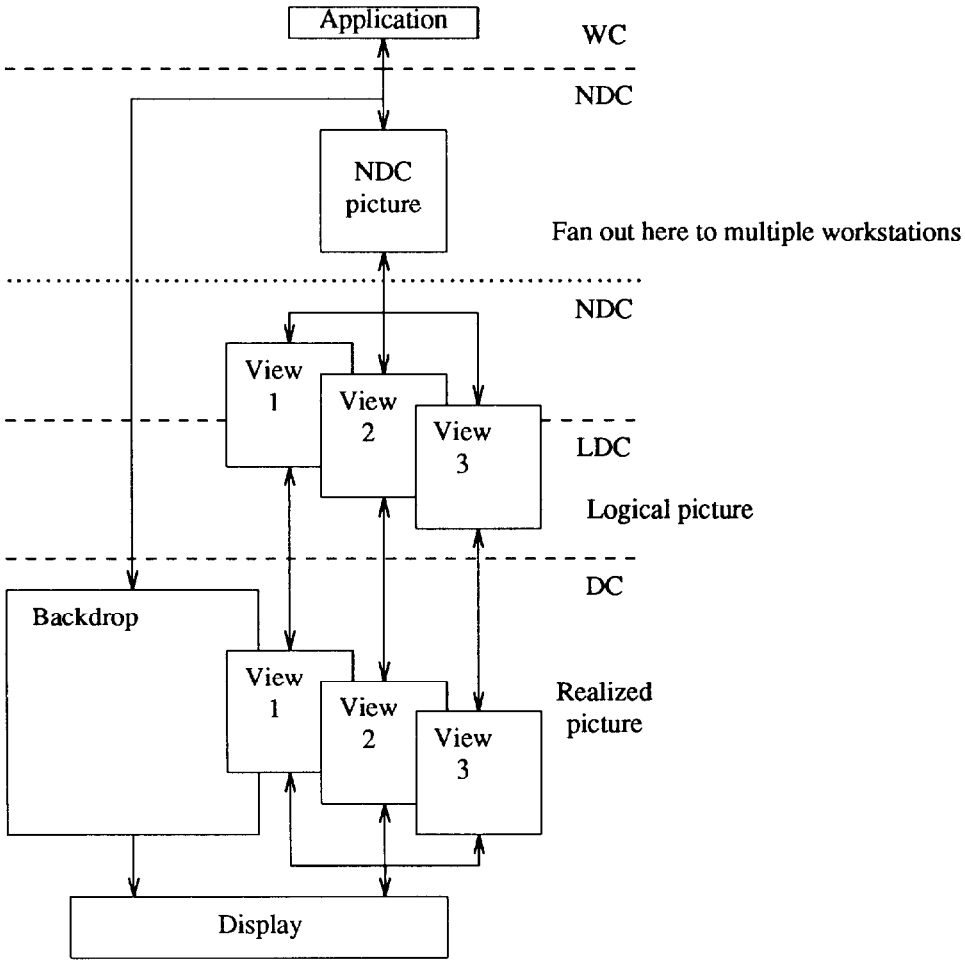

Figure 2 - GKS overview

Documents
Document Sharing Hub
Full document details: ISO IEC 07942-1-1994

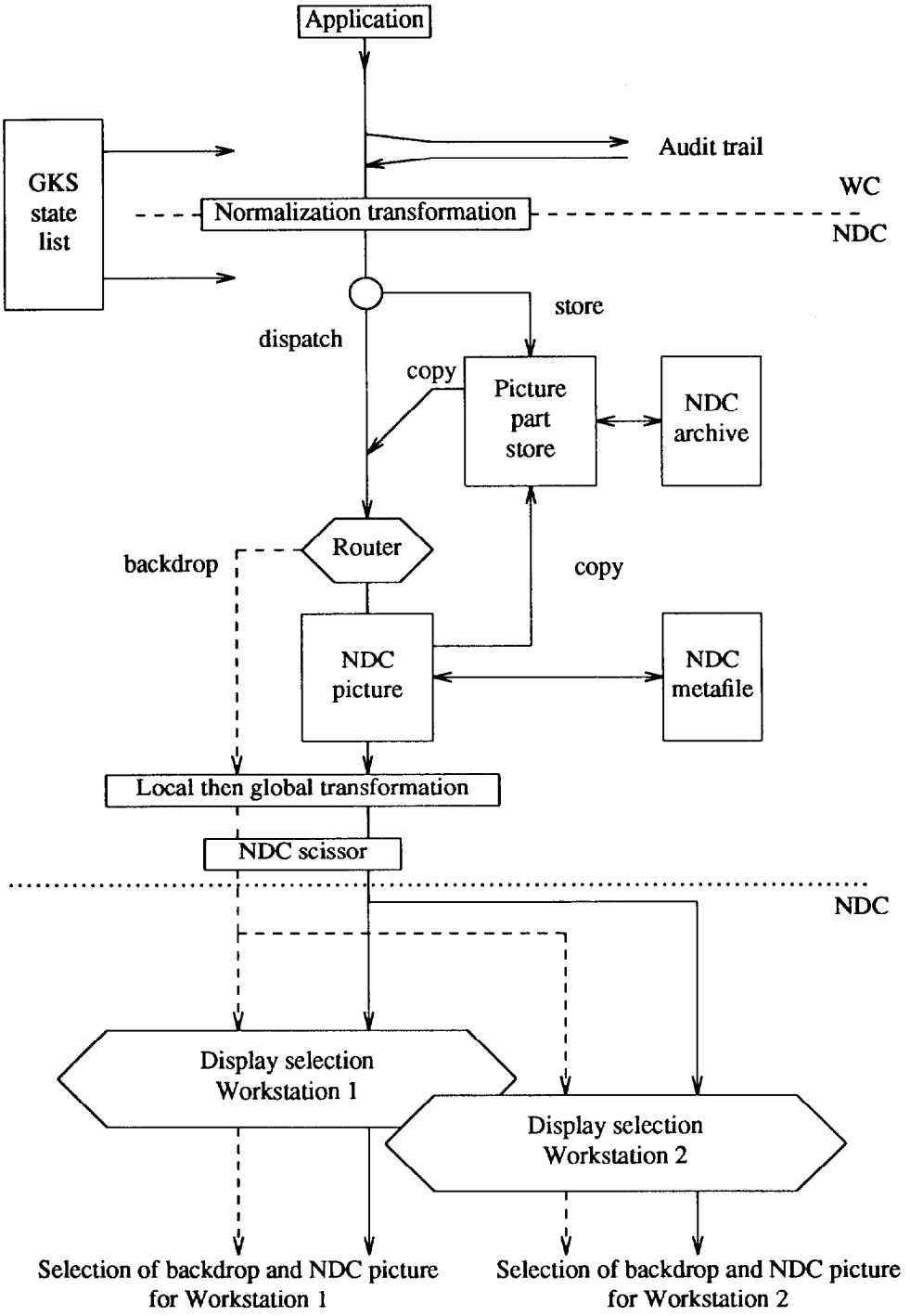

Figure 3 - GKS NDC picture

Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

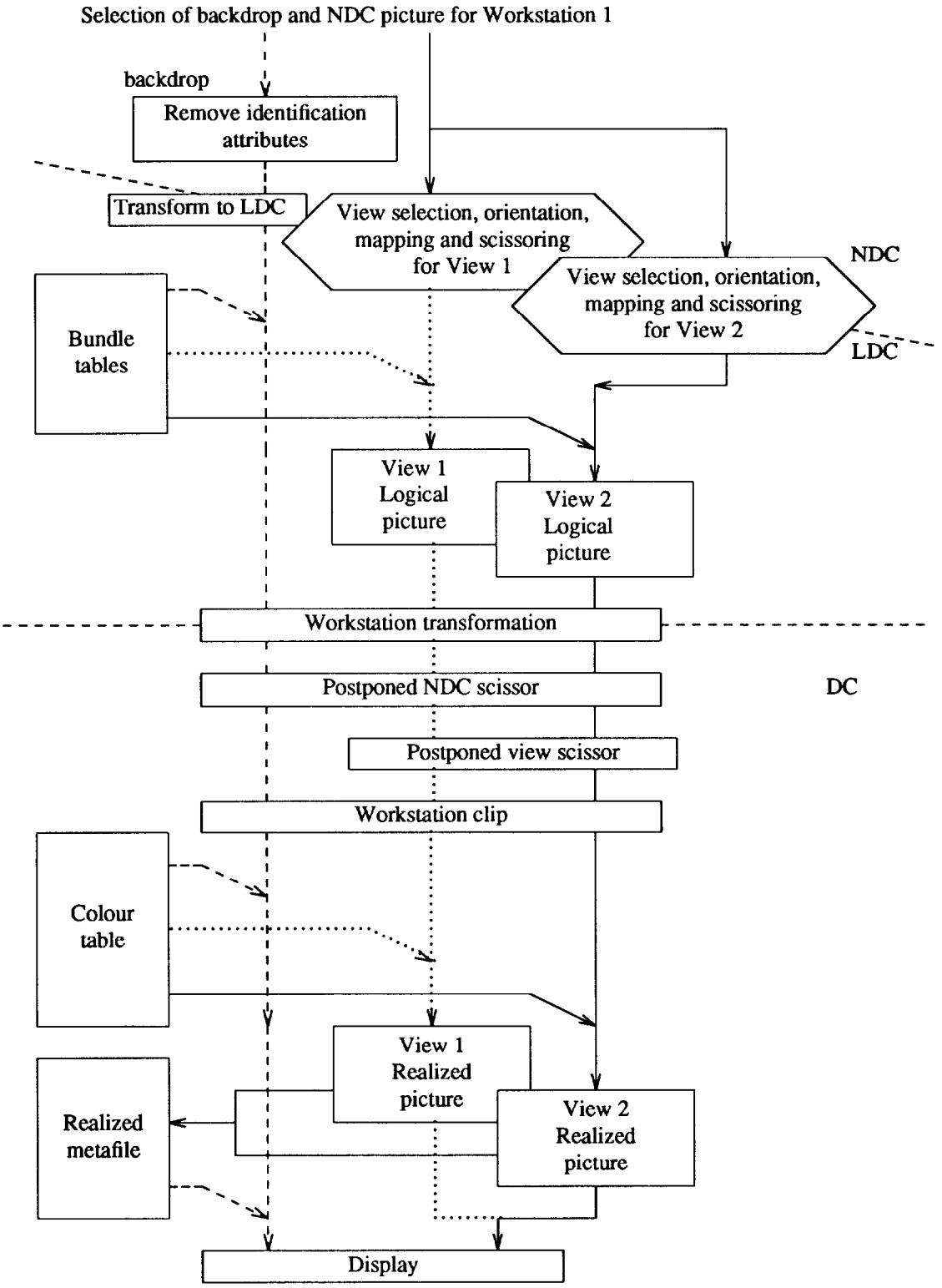

Figure 4 - GKS logical and realized pictures

Documents
Document Sharing Hub
Full document details: ISO IEC 07942-1-1994

5.4 Coordinate systems and transformations

Output primitives are defined by the application in a *world coordinate* system. The world coordinates are specified by the application. More than one world coordinate system may be specified.

The application specifies the transformation from world coordinates to *normalized device coordinates* (NDC). NDC is a workstation independent coordinate system.

The application specifies the orientation and mapping from NDC to LDC for each view on a workstation.

The display space of a workstation has a *device coordinate* system (DC) associated with it. The logical picture is defined in LDC and the realized picture in DC. The application specifies the mapping from LDC to DC for each workstation.

An output primitive and its NDC attributes are mapped from WC to NDC by a *normalization transformation*. The resulting geometry of an output primitive in NDC space is transformed by the local transformation and then by the global transformation. The output primitive may appear in several views on a workstation with different orientation and mapping from NDC to LDC. The workstation transformation performs the mapping from LDC to DC.

5.5 Logical input devices

A *logical input device* is an abstraction of one or more physical devices. An application can define how it receives *logical input values* from logical input devices.

Logical input devices are divided into classes dependent on the datatype of the logical input value. All logical input devices of one class deliver logical input values of the same datatype.

Three operating modes are defined for every logical input device. The operating modes specify whether the operator or the application has the initiative in controlling input.

5.6 Picture part store

Sequences of output primitives with associated attributes may be defined as a *picture part* and retained in *picture part store*. Positions associated with output primitives and attributes are stored in NDC coordinates. The NDC picture and the backdrop may be augmented by adding the sequence of output primitives in a picture part. The global and local transformation attributes of output primitives in a picture part may be changed as the picture part is added to the NDC picture, thus enabling several instances of a picture part to appear in the NDC picture. The namesets of output primitives in a picture part may be changed as the picture part is added to the NDC picture. Picture parts may also be created by selecting output primitives from the NDC picture.

5.7 State lists

State lists precisely describe the current state of an application in its use of graphics. Two types of state list exist. The *GKS state list* describes the current state of the NDC environment as shown in figure 3 and the input queue. The *workstation state list* describes the current state of a workstation. Several workstation state lists may be in use at the same time.

5.8 Description tables

The details of a particular GKS implementation are defined in a set of description tables. The *GKS description table* contains information about the specific implementation of GKS. A *generic workstation description table* defines the characteristics of a type of workstation and a *specific workstation description table* defines the characteristics of an instance of a generic workstation type.

5.9 Metafiles and archives

The contents of the NDC picture and realized picture may be stored and retrieved from metafiles.

The picture part store may be archived.

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

5.10 Routing to the backdrop

A *backdrop* is defined. Output primitives may be *routed* to the backdrop or the NDC picture. Output primitives in the backdrop appear on the workstation but are not part of the NDC, logical, or realized pictures. Output primitives in the backdrop are rendered using the relevant NDC and logical attributes. On creation, an output primitive is routed to the backdrop and selected for display on workstations. Once the selection has been made, the identification attributes for primitives in the backdrop are removed so that it is not possible for the operator to interact with information displayed in the backdrop. Attributes are bound to the primitives before they are displayed on the workstations. Primitives in the backdrop are transformed from NDC to LDC by an identity transformation.

The realized picture and backdrop are conceptually dispatched to the display of the workstation on separate channels. Their relative positioning is display dependent. Anything that destroys the integrity of the backdrop will cause it to be deleted from the display (see 7.10).

5.11 Audit trail

Audits of the sequence of GKS functions invoked by an application may be captured selectively in *audit trails*. User records may also be added. Functions are provided to playback audits for inquiry and execution of the output functions. An audit can be used to store the GKS functions that define a backdrop for later playback.

5.12 Segments

Segment functionality is retained in this part of ISO/IEC 7942 for compatibility with ISO 7942:1985 (see 10.1). It is achieved by naming, picture parts and selection criteria which give a more powerful means of selecting data for display.

An open segment is defined as a collection of output primitives in the NDC picture identified by a mapping of the segment name. On closure, the output primitives that make up the segment are stored as a picture part in picture part store with a picture part name that is a mapping of the segment name. Segment attributes are specified by selection criteria and primitive attributes.

5.13 Compatibility with ISO 7942:1985

Each GKS implementation shall ensure that it is possible to provide the functionality of ISO 7942:1985. Annex G illustrates one way of achieving it.

A GKS implementation has to support a special workstation type called WISS for compatibility with ISO 7942:1985. A WISS workstation is handled as in ISO 7942:1985 as far as state list entries are concerned, and for segment operations.

It is anticipated that this compatibility requirement will not be required for the next edition of GKS.

5.14 Summary

The main concepts are all contained in figures 3 and 4. Tables specific to single primitive classes have not been shown. Figure 3 shows the application accessing GKS (through a language binding). The commands generated by the application can be stored in an audit trail and replayed later. The commands can change values in the GKS state list (define text height, select normalization transformation etc) and these are either stored in world or NDC coordinates depending on the specific item. This is why the GKS state list straddles the normalization transformation. The geometric information defining the output primitive is transformed to NDC by the selected normalization transformation and attribute values are bound to the primitive from the GKS state list. Output primitives can either be dispatched for output or stored as a part of a picture part in picture part store. For long term storage (between sessions), it is possible to archive some or all of the picture part store.

Output consists of output primitives either generated by the application or copied from picture part store. The router directs these output primitives to the backdrop or to the NDC picture. The backdrop is provided so that large background pictures can be output without the need for local storage. In general, output primitives will

AVA Documents Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

be added to the NDC picture. These will have attributes associated with them defining both a local and a global transformation and the NDC scissor to be applied. The output primitive in the NDC picture selected for display will have its geometry transformed first by the local transformation and then by the global transformation before being scissored by the NDC scissor. The local transformation can be used in the creation of objects while the global transformation is usually used for the positioning of objects made up of a number of output primitives. The NDC scissor may be postponed until after the workstation transformation if scissoring of the output primitives' shape rather than locus is required.

The NDC picture can be captured in an NDC metafile.

A selection criterion on the identification attributes of the output primitives in the NDC picture or backdrop defines which output primitives will be displayed on each workstation.

Figure 4 shows what happens to the set of output primitives selected for display on a specific workstation. Each output primitive selected from the NDC picture can appear in several views on a workstation. A second selection criterion selects the set of output primitives to appear in each view. For each view, the output primitives can be reorientated and mapped to a different position in LDC space. The values of logical attributes (from the bundle tables) may replace the values bound in the NDC picture, depending on the values of the source attributes, to create the logical picture.

The backdrop has the identification attributes of its output primitives removed to ensure that the operator cannot interact with the backdrop. No view selection takes place. As can be seen by the dashed lines emanating from the bundle table and colour table, the backdrop has the relevant bundle table information bound to it in the same way that the output primitives from the NDC picture have the information bound to create the logical picture. An identity NDC to LDC transformation is applied. The backdrop has the workstation transformation and clip applied and the colour information for indirectly specified colours is extracted from the colour table for primitives for which colour is specified indirectly.

The logical picture in LDC coordinates is transformed and clipped to produce the realized picture in DC coordinates. It is possible to postpone both the NDC scissor and view scissor to take place after the transformation of the logical picture and before workstation clipping. At this stage, colour information for indirectly specified colours is extracted from the colour table so that the realized picture is completely defined ready for display.

It is possible to store the realized picture as a sequence of output primitives in the realized metafile. Realized metafiles can be read back into the backdrop.

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

6 The Graphical Kernel System

6.1 Initialization

To activate GKS, the function OPEN GKS is invoked. This defines a set of data structures (called *state lists* and *description tables*) which define the characteristics of the implementation of GKS in use. State lists are dynamic and may change during program execution. Description tables define the characteristics of a particular system. The following data structures are initialized on GKS being opened:

a) Operating state list: defines the state of GKS.

b) GKS description table: gives information about the workstations and fonts available.

c) GKS state list: provides information about the state of GKS as the execution of a program progresses. On OPEN GKS being invoked, it is set up with predefined default values.

d) Generic workstation description tables: these each describe the characteristics of a type of workstation. One is provided for each type in the GKS implementation accessed.

Inquiry functions are provided to access all the information in the data structures. The functions SAVE GKS STATE LIST and RESTORE GKS STATE LIST allow parts of the GKS state list to be saved by the application and restored.

To deactivate GKS, the function CLOSE GKS is invoked. The data structures listed above no longer exist.

6.2 Graphical output

6.2.1 Output primitive classes

Output primitives are generated by GKS. Output primitives may be stored in the picture part store, or dispatched to the router for routing to the NDC picture or backdrop.

The basic geometry of the output primitive is part of its definition. Additional properties of an output primitive are defined by primitive *attributes* as described in 6.2.2.

Output primitives are divided into the following output primitive classes:

a)	CURVE:	generates a set of curves
b)	MARKER:	generates a set of symbols
c)	AREA:	generates a set of areas with edges
d)	CHARACTER:	generates a sequence of glyphs
e)	IMAGE:	generates an array of cells with individual colours
f)	DESIGN:	generates graphical output by extruding a tiling through a stencil

There is one kind of output primitive which does not belong to any particular class called the *generalized drawing primitive* (GDP) (see 8.8).

6.2.2 Output primitive attributes

Output primitives have identification, NDC, source and logical attributes. All the relevant primitive attributes (see 11.1.5) are bound to an output primitive when it is created by invoking the function CREATE OUTPUT PRIMITIVE. For the backdrop, identification attributes are removed after output primitives have been selected for display on a workstation so that it is not possible for the operator to interact with the backdrop. The function SET PRIMITIVE ATTRIBUTE defines all the attribute values in the GKS state list that may be bound to a primitive on creation. One value is set at a time by specifying the attribute and the new attribute value to be associated with it. This value is bound on creation to all subsequent primitives to which it refers until the value is reset by another invocation of SET PRIMITIVE ATTRIBUTE for this specific attribute. All

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

attributes of output primitives in the current NDC picture may be changed by the function SET NDC PICTURE PRIMITIVE ATTRIBUTE.

All output primitives have the *nameset* and *pick identifier* identification attributes associated with them.

Output primitives that make up the current NDC picture can have their NAMESET attributes changed by invoking one of the functions ADD SET OF NAMES TO NDC PICTURE, or REMOVE SET OF NAMES FROM NDC PICTURE. The functions ADD SET OF NAMES TO NAMESET and REMOVE SET OF NAMES FROM NAMESET allow the current nameset value in the GKS state list to be modified.

The PICK IDENTIFIER attribute provides an additional level of identification used in input (see 9.2).

All output primitives have the following three NDC attributes associated with them:

a) LOCAL TRANSFORMATION

b) GLOBAL TRANSFORMATION

c) SCISSOR SET

For some classes of output primitives, additional NDC attributes are defined. NDC attributes complete the geometric definition of output primitives.

The geometry of the primitives is transformed by the LOCAL TRANSFORMATION matrix and then the GLOBAL TRANSFORMATION matrix before NDC scissoring takes place. The two transformations are defined with respect to the NDC coordinate space. Global transformations allow transformations to be applied to complete objects while local transformations can be used to position individual output primitives.

Scissoring of the NDC picture restricts the appearance of each output primitive either by clipping or shielding. NDC scissoring is specified by the SCISSOR SET attribute which consists of a set of named scissors. Scissors are named by scissor identifiers. A scissor consists of a clipping indicator, clipping rectangle set, shielding indicator and shielding rectangle set. Clipping and shielding rectangles are defined in NDC. Clipping and shielding are activated by the clipping indicator and shielding indicator respectively. When NDC scissoring takes place depends on the scissor mode set by invoking the SET SCISSOR MODE function. If scissor mode is set to LOCUS, NDC scissoring occurs after the local and global transformations have been applied and, conceptually, before the output primitive is sent to the relevant workstations. A part of an output primitive is visible if its locus is outside all shield rectangles and inside at least one clip rectangle. If the clip rectangle set is empty and the clip indicator is CLIP, all of an output primitive is visible. If scissor mode is set to SHAPE, NDC scissoring is postponed until after the workstation transformation. Any relevant transformations are applied to the NDC scissor so that its effect, if postponed, is the same as when it is applied immediately except for the differences due to scissoring on shape rather than locus.

The functions ADD SET OF SCISSORS TO NDC PICTURE and REMOVE SET OF SCISSORS FROM NDC PICTURE allow a specified set of scissors to be added to or removed from the SCISSOR SET attribute of a group of primitives in the NDC picture. The function SET NDC PICTURE PRIMITIVE ATTRIBUTE can be used to replace the SCISSOR SET attribute bound to a group of primitives.

The functions ADD SET OF SCISSORS TO SCISSOR SET and REMOVE SET OF SCISSORS FROM SCISSOR SET allow the current scissor set attribute value in the GKS state list to be modified.

Current values of NDC attributes associated with specific primitives are defined in world coordinates and are stored in the GKS state list in world coordinates. When these NDC attributes are bound to their respective primitives, the values are subject to the same transformations as the geometric data contained in the definition of the primitive.

Source attributes are bound to an output primitive when the primitive is created. Each class of primitive has two source attributes, a set of ATTRIBUTE SOURCE FLAGS (ASFs) which determine the source of logical attributes which are applied in the logical picture, and a bundle index for that class of primitive. Each bundle index value is an index into a bundle table for that class of primitive, which exists on each workstation (stored in the workstation state list). The values in a particular bundle (or entry in a bundle table) may be different for different workstations. The ASFs control whether the values of the logical attributes of a primitive at the logical level are taken from the values bound to the primitive at the NDC level or from the bundle table. A full

Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

description of logical attributes is given in clause 8.

6.3 Normalization transformations

The application programmer can compose a graphical picture from separate entities each of which, conceptually, is defined with its own world coordinate system (WC). The relative positioning of the separate entities is defined by having a single normalized device coordinate space (NDC) onto which all the defined world coordinate systems are mapped. A set of normalization transformations defines the mappings from the world coordinate systems onto the single normalized device coordinate space, which can be regarded as a workstation independent abstract viewing space. On creation, output primitives have world coordinate positions transformed to NDC coordinates before being dispatched to the router or stored in the picture part store. A single normalization transformation is current at any one time and this is used for the transformation from WC to NDC.

A normalization transformation is specified by invoking the function SET WINDOW AND VIEWPORT which defines the limits of an area in the world coordinate system (window) which is to be mapped onto a specified area of the normalized device coordinate space (viewport). Window and viewport limits specify rectangles parallel to the coordinate axes in WC and NDC (see figure 5). The rectangles include their boundaries. The normalization transformation performs a mapping from WC onto NDC that includes translation and differential scaling with positive scale factors for the two axes.

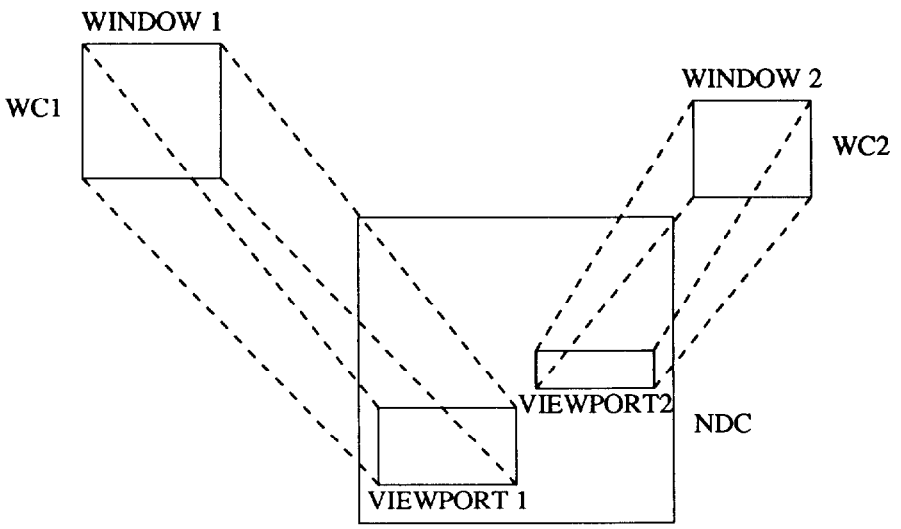

Figure 5 - Normalization transformations

Although NDC space conceptually extends to infinity, an implementation may support only a restricted range of NDC. Coordinates in the closed range [-7,7] × [-7,7] are always handled.

Each normalization transformation is identified by a *normalization transformation number* which is a non-negative integer. The normalization transformation with normalization transformation number 0 is the identity transformation which maps [0,1] × [0,1] in world coordinates to [0,1] × [0,1] in normalized device coordinates. It cannot be changed.

Initially, all other normalization transformations are set to a default transformation which is the same as normalization transformation number 0. Different transformations can be specified at any time when GKS is open. Since GKS provides a number of different normalization transformations, it is possible for the application program to specify them prior to outputting the graphical picture. The separate entities in the picture are output by selecting a particular normalization transformation before outputting the associated graphical

AVA Documents Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

primitives. However, specifying a normalization transformation while the graphical output is taking place is allowed.

A normalization transformation can be selected by SET NORMALIZATION TRANSFORMATION NUMBER, and it will be used for all output until another is selected. By default, normalization transformation number 0 is selected.

6.4 Picture part store

6.4.1 Picture part creation

Output primitives on creation may be stored in a picture part or dispatched to the router. When an OPEN PICTURE PART function is invoked, primitives are stored in the picture part and dispatching ceases. When CLOSE PICTURE PART is invoked, the sequence of primitives that make up the picture part is added to the picture part store. Subsequent primitives are dispatched to the router. The picture part is identified by a unique, application specified picture part name. Primitives in the picture part have attributes bound to them in the same way as primitives dispatched to the router.

6.4.2 Picture part functions

The following functions are provided to manipulate picture parts:

a) RENAME PICTURE PART: changes the name of the picture part specified to a new name.

b) DELETE PICTURE PART: removes a picture part from the picture part store.

c) REOPEN PICTURE PART: reopens a picture part. Output primitives created are added in sequence to the end of the specified picture part until CLOSE PICTURE PART is invoked.

d) APPEND PICTURE PART: adds one picture part to the end of the other picture part. The output primitives can be altered in a similar way to COPY PICTURE PART FROM PICTURE PART STORE (6.4.3).

6.4.3 Adding picture parts

A picture part can be dispatched to the router by invoking COPY PICTURE PART FROM PICTURE PART STORE. In sequence, each output primitive in the picture part has its global and local transformation attributes replaced, or pre or post concatenated by the specified transformation matrices, and the specified set of names is added to the NAMESET attribute already associated with the output primitive. This allows several instances of a picture part to appear in the NDC picture in different positions and these can be differentiated by their primitives having different NAMESET attributes. The ability to change or reset the global and local transformation attributes allows the incoming primitives to be consistent with the primitives already in the NDC picture. Depending on a 'scissor select' parameter, either the current scissor set is bound to the primitives copied or the scissor set bound to the primitives in the picture part store is used.

Utility functions (EVALUATE TRANSFORMATION MATRIX and ACCUMULATE TRANSFORMATION MATRIX) are available to assist the application in establishing a transformation matrix.

6.4.4 Picture part archiving

A picture part can be archived by invoking the function ARCHIVE PICTURE PART. Each archive consists of a set of picture parts with names local to the archive. On archiving, an archive name is supplied. A picture part can be retrieved from the archive by invoking the function RETRIEVE PICTURE PART FROM ARCHIVE. On retrieval, a picture part name must be supplied. An error occurs if there is a conflict between an existing name in the archive or picture part store and the name supplied.

AVA Documents Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

6.5 The NDC picture

6.5.1 Introduction

The NDC picture at any time consists of a sequence of output primitives which have been added to the picture in one of three ways:

a) on creation outside the definition of a picture part;

b) by copying from a picture part;

c) on input from an NDC metafile.

The order of appearance of output primitives in the NDC picture is important in that if two output primitives overlap, the second primitive in the sequence may obscure some part of the first.

Each workstation is responsible for selecting the subset of the NDC picture to be viewed and rendering it.

6.5.2 NDC picture operations

The following operations can be applied to the NDC picture:

a) DELETE PRIMITIVES FROM NDC PICTURE: the primitives whose NAMESET attribute in the NDC picture satisfies the specified selection criterion are deleted from the NDC picture.

b) REMOVE SET OF NAMES FROM NDC PICTURE: the specified names are removed from the NAMESET attributes of all primitives in the NDC picture which satisfy the specified selection criterion.

c) ADD SET OF NAMES TO NDC PICTURE: the specified names are added to the NAMESET attributes of all primitives in the NDC picture which satisfy the specified selection criterion.

d) SET NDC PICTURE PRIMITIVE ATTRIBUTE: the primitives satisfying the specified selection criterion have the value of the specified attribute changed.

e) REMOVE SET OF SCISSORS FROM NDC PICTURE: the specified scissors are removed from the SCISSOR SET attributes of all primitives in the NDC picture which satisfy the specified selection criterion.

f) ADD SET OF SCISSORS TO NDC PICTURE: the specified scissor set is added to the SCISSOR SET attributes of all primitives in the NDC picture which satisfy the specified selection criterion.

g) COPY NDC PICTURE TO PICTURE PART STORE: the primitives satisfying the specified selection criterion are copied to the picture part store as the specified picture part. The NAMESET attribute associated with each primitive has the specified set of names removed.

h) REORDER NDC PICTURE: The subsequence of primitives satisfying the source selection criterion are moved in front of the first position in the remaining subsequence of primitives satisfying the reference selection criterion (BACK) or after the last position in the remaining subsequence of primitives satisfying the reference selection criterion (FRONT). The relative order of primitives which satisfy the selection criterion is not changed by this operation.

6.5.3 NDC metafiles

It is useful to be able to capture a subsequence of primitives of the NDC picture satisfying a selection criterion, and store it away for future use or transmission to other systems. The function COPY NDC PICTURE TO NDC METAFILE will store a subsequence of the NDC picture as a picture on the specified metafile adding a set of names to the NAMESET attribute of each primitive of the subsequence. The picture can be recovered at a later time and added to the current NDC picture by invoking COPY NDC METAFILE PICTURE TO NDC PICTURE. A set of names may be added to the NAMESET attribute of each primitive recovered. Adding a set of names to the NAMESET attribute allows the primitives captured or recovered to be identified. The backdrop cannot be sent to an NDC metafile.

AVA Documents Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

6.6 Selection criterion

Functions in GKS select subsequences of output primitives (for example, subsequences of the NDC picture or of a picture part) according to a *selection criterion*. A typical function of this type is ADD SET OF NAMES TO NDC PICTURE. A primitive will be included in the subsequence if its NAMESET attribute satisfies the selection criterion.

Selection criteria are constructed from comparison operations and logical operations. The comparison operations allowed are:

a) CONTAINS

b) ISIN

c) EQUALS

d) SELECTALL

e) REJECTALL

The first three operations specify a nameset which is compared with the nameset of each output primitive in the sequence.

All namesets satisfy the SELECTALL selection criterion and no namesets satisfy the REJECTALL criterion. These selection criteria can thus be used to select or reject all primitives in a sequence respectively.

More than one comparison operation can be included in the selection criterion by using the following logical operations:

f) AND

g) OR

h) NOT

The selection criterion CONTAINS (RED,GREEN), for example, will select all output primitives having namesets containing both names RED and GREEN. The selection criterion CONTAINS (RED,GREEN) OR CONTAINS (BLUE,YELLOW) will select those primitives with namesets containing either RED and GREEN or BLUE and YELLOW.

The selection criterion ISIN (RED,GREEN) will select those primitives whose namesets are subsets of the set (RED,GREEN). In this case that is primitives with the namesets (RED), (GREEN), (RED,GREEN) and the empty nameset.

A precise definition of selection criteria is given in 11.1.18. The sequence of primitives from which a selection is to be made are tested one by one against the selection criterion and, if accepted, are included in the resulting sequence and, if rejected, are excluded.

6.7 Graphical input

6.7.1 Introduction to logical input devices

An application program obtains graphical input from an operator by controlling the activity of one or more logical input devices, which deliver logical input values to the program.

The *device identifier* defines the class of logical input device and how it relates to physical devices on a workstation. The logical input device class determines the type of logical input value delivered. The classes and the logical input values they provide are:

a) LOCATOR: a position in world coordinates and a normalization transformation number.

b) STROKE: a sequence of points in world coordinates and a normalization transformation number.

AVA Documents Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

c) VALUATOR: a real number.

d) CHOICE: a CHOICE status and a positive integer which represents a selection from a number of choices.

e) PICK: a PICK status, a nameset and a pick identifier.

f) STRING: a character string.

g) COMPOSITE: an application defined logical input device which delivers values that are composites of values of the above types.

Each logical input device may be operated in three modes, called *operating modes*. At any time a logical input device is in one, and only one, of the modes set by the invocation of the function SET LOGICAL INPUT DEVICE MODE. The three operating modes are REQUEST, SAMPLE and EVENT. Input from devices is obtained in different ways depending on the mode as follows.

h) REQUEST: A specific invocation of REQUEST INPUT causes an attempt to read a logical input value from a specified logical input device. This can only occur when the logical input device is in REQUEST mode. GKS waits until the input is entered by the operator or a break action is performed by the operator. The break action is dependent on the logical input device and on the implementation. If a break occurs, the logical input value is not valid.

i) SAMPLE: A specific invocation of SAMPLE INPUT causes GKS, without waiting for an operator action, to return the current logical input value of a specified logical input device. This can only occur when the logical input device is in SAMPLE mode.

j) EVENT: GKS maintains one input queue containing temporally ordered event reports. An event report contains the identification of a logical input device and a logical input value from that device. Event reports are generated asynchronously, by operator action only, from input devices in EVENT mode.

The application program can remove the oldest event report from the queue and examine its contents. The application can also flush from the queue all event reports from a specified logical input device.

A specific logical input device is said to be taking part in an interaction during the whole time that it is in SAMPLE or EVENT mode, but, when it is in REQUEST mode, only during the execution of a REQUEST INPUT function for that device. Alternatively, an interaction with the device can be said to be underway during that time. Many devices on many workstations may be taking part in interactions simultaneously.

6.7.2 Logical input device model

To describe the precise actions of the logical input devices, it is first necessary to describe their relationship with physical input devices, using the concept of measures and triggers.

A logical input device contains a measure, a trigger, an initial value, a prompt and echo type, an echo area and a data record containing details about the prompt and echo type. A logical input device's measure and trigger are parts of the implementation of the workstation containing the logical input device. Initial value, prompt and echo type, echo area, and data record can be supplied by the application program.

The measure of a logical input device is a value determined by one or more physical input devices together with a 'measure mapping'. More than one measure may simultaneously be determined by a single physical device; a separate measure mapping applies for each measure. A measure can be seen as the state of an independent, active process (a measure process). Each state corresponds exactly with a logical input value.

The current state of the measure process (i.e. the device's measure) is available as a logical input value. Whenever the device is taking part in an interaction, the measure process is in existence. Under other conditions, this process does not exist.

AVA Documents Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

While the measure process exists, if echoing is required, information indicating the current state of the measure process is provided to the operator.

The trigger of a logical input device is a physical input device or a set of them together with a 'trigger mapping'. The operator can use a trigger to indicate significant moments in time. At these moments, the trigger is said to 'fire'. A single operator action (for example, pressing a button) causes the firing of not more than one trigger. Several logical input devices may refer to the same trigger.

A trigger can be seen as an independent, active process (a trigger process) that sends a message to one or more recipients when it fires. A logical input device is a recipient of its trigger if there is a pending REQUEST for it or if it is in EVENT mode. Both of these conditions can be true simultaneously for different logical input devices. If there is at least one recipient for a trigger, the trigger process is in existence. Under other conditions this process does not exist.

If a REQUEST for a logical input device is pending when the device's trigger fires, the measure of that device is used to satisfy the REQUEST (see figure 6). If one or more devices containing a given trigger are in EVENT mode when the trigger fires, the identifications of those devices and their measure values are passed to the input queue mechanism as separate event reports. The input queue mechanism is described in detail in 6.7.4.

When a trigger firing succeeds in satisfying a REQUEST, or adding event records to the input queue, GKS provides to the operator an acknowledgement the form of which depends on the implementation of the logical input device. The acknowledgement is not controllable by a GKS function.

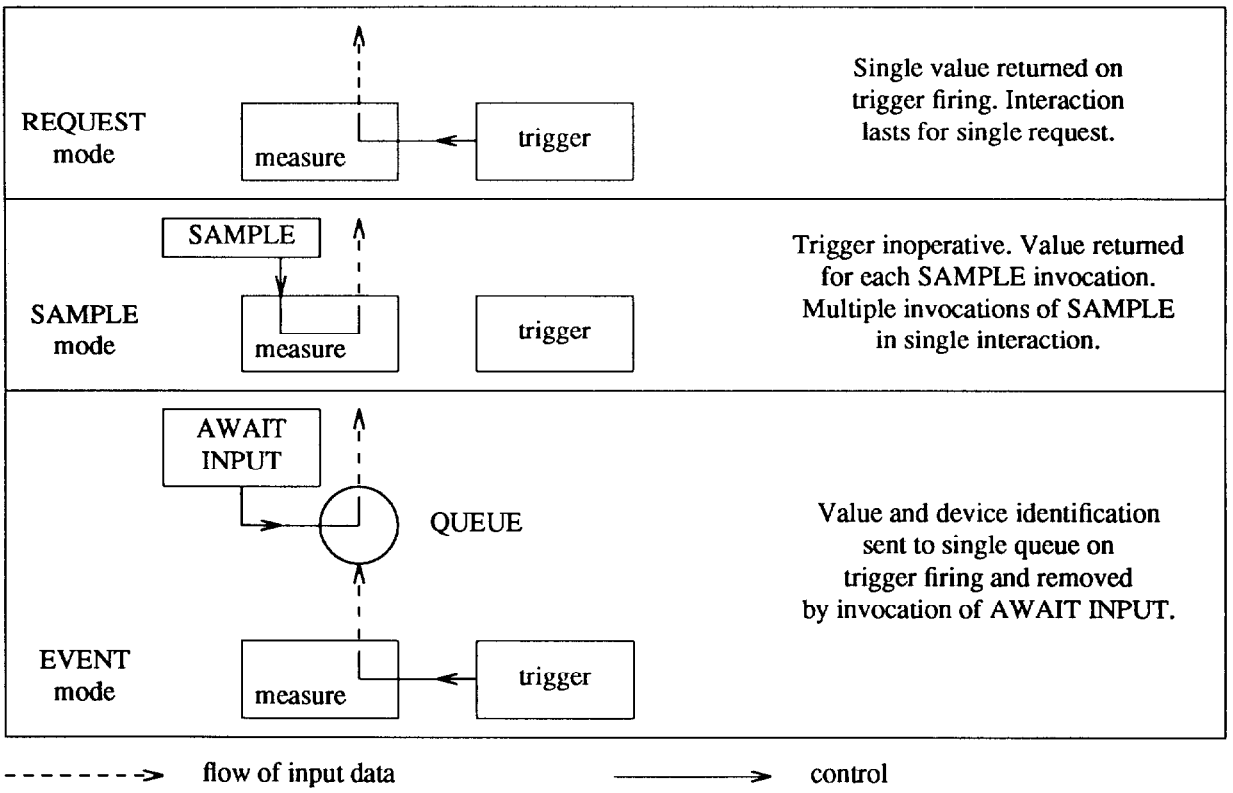

Figure 6 - Operating modes using measures and triggers

Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

6.7.3 Operating modes of logical input devices

The mode of a logical input device can be changed by invoking the appropriate SET LOGICAL INPUT DEVICE MODE function.

After an invocation of SET LOGICAL INPUT DEVICE MODE with the parameter REQUEST, no measure process exists for the specified device and the device's identifier is not on its trigger's list of recipients. After an invocation with the parameter EVENT, a newly initiated measure process is in existence for the specified device and the device's identifier is on its trigger's list of recipients. After an invocation with the parameter SAMPLE, a newly initiated measure process is in existence for the specified device, but the device's identifier is not on its trigger's list of recipients.

Initially a logical input device is in REQUEST mode. While a device is in REQUEST mode, a logical input value can be obtained by invoking the appropriate REQUEST INPUT function. The effects of doing so are:

a) To create a measure process for the specified device and set its initial value. Echoing is performed by the measure process if echoing is on for the specified device.

b) To add the device's identifier to its trigger's list of recipients. If the list was previously empty, the trigger process is started.

c) To suspend GKS until the trigger of the specified device fires, or the operator invokes the break facility.

d) If the trigger fired, to set the logical input value to the current state of the measure process.

e) To destroy the measure process.

f) To remove the device's identifier from its trigger's list of recipients. If this list becomes empty, the trigger process is destroyed.

g) If the trigger fired, to return the logical input value and the status OK, otherwise to return the status NONE.

While a logical input device is in SAMPLE mode, a logical input value can be obtained by invoking the SAMPLE INPUT function. The effect of doing so is to set the logical input value to the current state of the measure process without waiting for a trigger firing.

While a logical input device is in EVENT mode, logical input values are added as event reports to the input queue, and can be obtained in sequence by invoking AWAIT INPUT. When an event report is generated, the state of the measure process depends on the input device class. For all classes other than STRING (and any STRING component of a COMPOSITE class), the state of the measure process is unchanged. For STRING class input, it is reset to its initial value.

Figure 6 shows the effect of every operating mode on the measure and trigger of a logical input device.

6.7.4 Input queue and current event report

The input queue contains zero or more event reports. Event reports contain pairs of values (device identifier, logical input value) resulting from trigger firings. Event reports may be added to the input queue when logical input devices in EVENT mode are triggered by the operator. Events can be removed from the input queue by invocations of AWAIT INPUT (see figure 6) and FLUSH DEVICE EVENTS.

When a trigger that is part of one or more logical input devices in EVENT mode fires, the resulting event reports are entered into the queue. A set of event reports, one for each device is added to the input queue, if and only if there is room for the whole set of simultaneous event reports. AWAIT INPUT returns all the simultaneous event reports together.

If there is not room in the queue for all event reports when a trigger fires, input queue overflow has occurred. Input queue overflow is not reported to the application program immediately. It is reported via the error mechanism during the next invocation of any GKS function that can remove event reports from the input queue (AWAIT INPUT, FLUSH DEVICE EVENTS, and CLOSE WORKSTATION (see 7.3)). The input queue has to be emptied before further event reports will be added. Between the detection of input queue overflow and the next time AWAIT INPUT is invoked with the input queue empty, no events are generated by trigger firings and thus no acknowledgements are provided. (This permits the application program to

AVA Documents Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

determine how many events were in the queue when overflow occurred by calling AWAIT INPUT with zero timeout.)

When the 'input queue overflow' error is reported, the trigger causing the overflow is indicated by placing into the error state list the identification of any one of the logical input devices using that trigger which was in EVENT mode at the time the overflow was detected.

AWAIT INPUT, if the queue is not empty, removes the first event report and returns it to the application program. If the queue is empty, AWAIT INPUT suspends execution until an event report is queued or until the specified timeout period has elapsed.

FLUSH DEVICE EVENTS removes all event reports for a specific device from the input queue.

6.8 Inquiry functions

Inquiry functions return values directly from or derived from the various state lists and description tables. The data types of the values and the default values of the entries are summarized in clause 11.

The inquiry functions are designed in such a way that they do not cause any errors to be generated. Inquiry functions for values that may be logically unavailable have an output parameter, 'error indicator', that determines whether or not the other returned values are valid. The availability parameter is of the type integer and, in the event of the other values not being available, returns an error number, which identifies the appropriate GKS error condition. The same error numbers are used for inquiry functions as for other functions and thus the standard list of error messages should be consulted. If GKS is not in the proper state, then the error number appropriate to this condition is the one returned, even if there are other reasons for the values being unavailable. If the values are available, zero is returned in the error indicator parameter.

For all values except zero the returned output values are implementation dependent. The description of each inquiry function lists the error indicator values that the function may return.

6.9 Error handling

For each GKS function, a finite number of error situations is specified, any of which will cause the ERROR HANDLING function to be invoked. The ERROR HANDLING function defined invokes an ERROR LOGGING function which appends an error message and identification of the GKS function (which caused the error) to the error file before returning to the ERROR HANDLING function.

This two stage invocation of error handling allows the ERROR HANDLING function provided to be replaced by the application while still having access to services provided by the ERROR LOGGING function.

Each GKS function checks that it is used appropriately and that the values of input parameters are valid before attempting to execute the function. At least the first error detected will cause the ERROR HANDLING function to be invoked.

The application supplied ERROR HANDLING function is only allowed to invoke GKS inquiry functions, the ERROR LOGGING function and the EMERGENCY CLOSE GKS function. Inquiry functions are not allowed to generate errors.

If the application detects errors outside GKS and regains control, it can invoke the EMERGENCY CLOSE GKS function which will attempt to save as much of the graphical information as possible. GKS itself may invoke the EMERGENCY CLOSE GKS function if it gets into difficulty.

6.10 Special interfaces between GKS and the application program

A uniform escape mechanism for allowing access to installation and hardware specific features (a 'standard way of being non-standard') is provided by means of the ESCAPE function. Although the use of this mechanism reduces the portability of the application program, it does so in an easily identifiable manner.

The ESCAPE function does not generate graphical output; by contrast, the GENERALIZED DRAWING PRIMITIVE may generate graphical output not otherwise generated by GKS.

AVA Documents Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

6.11 Backdrop

By default, routing is to the NDC picture.

Invoking the function ROUTE with the parameter BACKDROP will cause output primitives to be routed directly to the backdrop. Invoking the function ROUTE with parameter NDC will cause output to be added to the NDC picture again.

Primitives, as they are routed to the backdrop, have their identification attributes removed after selection for display on a workstation and, therefore, cannot be picked.

6.12 Audit and playback

The function AUDIT can be used to control the capture of GKS functions executed by an application. The parameters to the AUDIT function are the name of the audit trail and the operation to be performed (which, for the OPEN operation, includes the specification of the connection to be established). Multiple audits are allowed simultaneously. The initialization operation OPEN should be performed before an audit is started. The operation BEGIN starts the recording of audit trail information and END stops it. BEGIN and END operations may be invoked several times during an audit so that a selection of the GKS functions invoked are recorded. The operation CLOSE stops the audit and releases the audit trail file.

For each GKS function executed, the function name and a list of the parameter values are stored in the set of open audits when recording is enabled. For functions, such as input and inquiry, both the parameter values input to the function and the output values generated by the function are recorded. Non-graphical information supplied by the user can also be added to the audit by the function WRITE USER RECORD TO AUDIT. This can be used to add information such as times of entries or the association of application specific data with the graphical information.

To playback a previously recorded audit, four functions are provided (PLAYBACK, READ ITEM FROM AUDIT, READ ITEM FUNCTION NAME FROM AUDIT, PROCESS AUDIT ITEM). The function PLAYBACK is used to OPEN and CLOSE the session for the specified audit trail replay. The information stored on the audit trail consists of a sequence of items, one per GKS function invoked. The function READ ITEM FROM AUDIT returns the name of the function executed and a list of parameter values associated with that execution of the function. The function READ ITEM FUNCTION NAME FROM AUDIT just returns the name of the function. The function PROCESS AUDIT ITEM identifies the audit and has a parameter which takes the values SKIP or DO. The SKIP operation changes the item pointed at to the next on the specified audit trail. Initially, the audit trail is positioned at the first item. The DO operation has a null effect for those functions not concerned with output. For functions related to output (described in 12.2, 12.3, 12.4, 12.5, 12.6, 12.7, 12.8, 12.10, 13.1 and 14), the effect is as if the GKS function was executed directly at this point with the parameters defined in the audit trail. The PROCESS AUDIT ITEM function changes the item pointed at to the next item in either case of the operation.

The function AUDIT invoked with the operation parameter BEGIN or END will cause the corresponding item to be added to the audit trail. These can be used to identify the individual parts of the overall audit trail.

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

7 Workstation dependent control

7.1 Introduction

When a workstation is opened, a selection criterion is established for the workstation which selects part of the NDC picture and backdrop for rendering on the workstation. For each output primitive, the local and global transformations are applied and then either the NDC scissor is applied before rendering takes place or it is postponed until after the workstation transformation. The operation of rendering the NDC picture on a workstation proceeds in three stages. In the first stage, the selection criterion for each view on the workstation is used to define how many views the output primitive will appear in. For each of these, the output primitive is first reorientated, then mapped to a new position and locus in LDC space; finally it is either scissored by the view scissor, or the view scissor is postponed until after the workstation transformation and the NDC scissor is applied. In the second stage, any values of logical attributes required from the workstation dependent bundle tables are bound and the resulting primitives are rendered to produce the logical picture (see figure 4). In the logical picture, colour is not resolved beyond colour specifier or colour array specifier. The third stage applies the workstation transformation (and if postponed, applies the NDC scissor followed by the view scissor) and workstation clip, and resolves colour specifiers (replacing colour indices by the colours defined in the colour table for the workstation) to produce the realized picture which is displayed on the workstation display space. The realized picture may be sent to a metafile. The updating of the logical and realized pictures is continuous. Any change to the NDC picture produces a change as soon as possible in the logical picture. The realized picture will have any colour or workstation transformation changes made as soon as possible also. However, updating of the logical and realized pictures may be suspended in order that a group of updates can be made to appear atomic.

Rendering the backdrop on a workstation is similar. The output primitives routed to the backdrop have any values of logical attributes required from the workstation dependent bundle tables bound to them. The resulting primitives are rendered and the workstation transformation and clip applied. Finally, the colour specifiers are resolved to produce the backdrop which is displayed. The backdrop cannot be sent to a metafile. The backdrop cannot be suspended.

7.2 Workstation characteristics

For every type of workstation present in a GKS implementation a generic workstation description table exists which describes the standard capabilities and characteristics of the workstation. On opening a workstation, a specific workstation description table is created for that workstation containing information derived from: the generic workstation description table, the device itself, the application through the workstation specific information provided with the workstation type, and possibly other implementation dependent sources. The content of the specific workstation description table may change at any time while the workstation is open. Such changes are generated by events external to GKS. The application program can inquire which generic capabilities are available before the workstation is open. The specific capabilities can be inquired while the workstation is open. The application program can adapt its behaviour dependent on this information. If capabilities are requested that a particular workstation does not provide, or are not yet available, a standard reaction is defined.

The function INQUIRE SPECIFIC WORKSTATION DESCRIPTION TABLE ENTRY can be used to inquire the value of a workstation description table entry whilst the workstation is open. There is an entry state flag associated with each workstation description table entry for each workstation, which takes the values CHANGED and UNCHANGED. Whenever an entry in a specific workstation description table changes, the corresponding flag is set to CHANGED (for example, in a windowing environment when, as a result of an operator action, the display is resized). The value of the flag associated with a particular entry in the specific workstation description table of an open workstation is returned by the function INQUIRE SPECIFIC WORKSTATION DESCRIPTION TABLE ENTRY STATE. The function RESET SPECIFIC

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

WORKSTATION DESCRIPTION TABLE ENTRY STATE enables the application program to reset any particular flag to UNCHANGED.

An abstract graphical workstation with maximum capabilities

a) has one rectangular and addressable display space;

b) permits the specification and use of smaller display spaces than the maximum while guaranteeing that no display image is generated outside the specified display space;

c) renders the NDC picture to produce the logical picture, using (possibly workstation dependent) logical attributes, and binds indirectly specified colour values to produce the realized picture;

d) has one or more logical input devices for each input class;

e) permits REQUEST, SAMPLE and EVENT type input;

f) allows independent setting of logical input devices in REQUEST, SAMPLE or EVENT mode.

Actual workstations may provide more capabilities than those listed in the workstation description table. These cannot be used by GKS. However, if the workstation itself provides sufficient intelligence, the additional capabilities can be accessed via the ESCAPE function or GENERALIZED DRAWING PRIMITIVE, or used locally by the workstation operator. As an example, if a workstation has two display surfaces, the operator can switch locally from one to the other without notifying GKS or the application program. More than one display space can be controlled by GKS only by defining a separate workstation for each display space.

7.3 Selecting a workstation

The application program references a workstation by means of a workstation identifier. Connection to a particular workstation is established by the function OPEN WORKSTATION, which associates the workstation identifier with a workstation specifier and workstation type. The current state of each open workstation is kept in a workstation state list. Specified entries in the workstation state list can be saved and restored by invoking the functions SAVE WORKSTATION STATE LIST and RESTORE WORKSTATION STATE LIST.

Connections are released by the function CLOSE WORKSTATION.

7.4 Selection criteria

Selection criteria are provided to control which primitives in the NDC picture are displayed on a workstation and the highlighting and detectability of the primitives displayed. When a selection criterion is changed, the sequence of primitives in the NDC picture is examined and those that satisfy the new selection criterion are rendered again to produce a new realized picture for display or storage. First, the display selection criterion selects the sequence of primitives for display. The highlighting and detectability selection criteria determine which of this sequence of primitives is highlighted or made detectable.

For output primitives routed to the backdrop, only the display selection criterion has any effect and selects primitives for display. They are not highlighted or made detectable.

Clause 10 describes the segment and workstation activation facilities which are explicitly defined in order to provide compatibility with ISO 7942:1985. These facilities make use of the nameset and selection criteria mechanisms and consequently it is necessary to reserve a part of each selection criterion for manipulation by GKS. This is explained in clause 10. The function SET WORKSTATION SELECTION CRITERION sets the application specifiable part of the current selection criterion, without modifying the system controlled part.

When a workstation is opened, the selection criteria for display, highlighting and detectability are each set to REJECTALL. Consequently, no primitives will be selected for display on a workstation, unless the application first defines an appropriate selection criterion.

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

7.5 Viewing

Each output primitive selected for display on a workstation may appear in one or many views on that workstation. Initially, view 0 has the default selection criterion of SELECTALL and all other views have the default selection criterion of REJECTALL. Thus, initially all output primitives selected for display appear in view 0 and no other view. The function SET VIEW SELECTION CRITERION can redefine the selection criterion for each view on the workstation.

The function SET VIEW defines the orientation, mapping and scissoring to be applied to each output primitive selected for the view. *View orientation* effectively moves the NDC coordinate system to one more appropriate for the set of output primitives to be viewed. *View mapping* provides a mapping of the set of output primitives to a new position in LDC space. Finally, the *view scissor* parameter provides an optional view scissor on the output primitives being viewed. View scissoring either occurs here or may be delayed until after the workstation transformation and any delayed NDC scissoring.

View 0 provides an identity mapping from NDC to LDC, empty clipping and shielding rectangle sets and clipping and shielding indicators with values NOCLIP and NOSHIELD respectively. These parameters of view 0 cannot be changed.

Two utilities are provided for the construction of the view orientation and mapping matrices. The function EVALUATE VIEW ORIENTATION MATRIX defines a matrix that will move the origin of the set of output primitives and redefine the Y direction. The function EVALUATE VIEW MAPPING MATRIX defines a matrix that performs a window to viewport mapping from NDC space to LDC space.

The NDC picture is displayed through views in order of view priority. The primitives satisfying the selection criterion of the lowest priority view are displayed first in temporal order. Primitives in views of higher priority are displayed after those of lower priority. View priority is set by the function SET VIEW PRIORITY. View priority also affects LOCATOR and STROKE input measures (see 9.2).

7.6 Workstation transformations

An application can select independently for each open workstation some part of the LDC space in the range $[0,1] \times [0,1]$ to be displayed somewhere in the workstation display space. A particular workstation transformation is a mapping from LDC space onto the device coordinates (DC) for that workstation. Thus picture composition can be achieved using the normalization transformations. View orientation and view mapping allow different views of the NDC picture to be assembled in the logical picture. The workstation transformation allows different areas of the logical picture to be viewed on different workstations. For example, a correctly scaled drawing could be output to a plotter and simultaneously some part of the drawing could be displayed on the surface of an interactive terminal.

The units of device coordinates are metres on a device capable of producing a precisely scaled image (for example, on most plotters) and appropriate workstation dependent units otherwise (for example, on a display unit with unknown monitor size). In either case the device coordinate system maps onto the display space in the following way:

a) the DC origin is at the bottom left corner of the display space;

b) the device coordinate units are related to the display space in such a way that a square in device coordinates appears as a square on the display surface;

c) x and y increase to the right and upwards respectively.

On some devices, device coordinate units do not coincide with the addressable units used by the display, for example if the addressable units do not satisfy the above conditions.

The current size of the display space in device coordinate units is recorded in the specific workstation description table (see 7.2).

A default workstation transformation is provided when the workstation is opened. The workstation transformation can be redefined at any time while the workstation is open. The workstation transformation is a uniform mapping from LDC onto DC and thus performs translation and *equal* scaling with a positive scale factor

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

for the two axes. A workstation transformation is specified by defining the limits of an area in the LDC system within the range [0,1] × [0,1] (workstation window) which is to be mapped onto a specified area of the device coordinate space (workstation viewport). Workstation window and workstation viewport limits specify rectangles parallel to the coordinate axes in LDC and DC. The rectangles include their boundaries. To ensure that no output outside the workstation window is displayed, GKS clips at the workstation window boundaries, and this clipping cannot be disabled. As the workstation window is defined to be somewhere in the LDC range [0,1] × [0,1], this ensures that the only part of LDC space that can be viewed on any workstation lies in the range [0,1] × [0,1]. If the workstation window and workstation viewport have different aspect ratios, the scaling specified would be different on each axis if the workstation window was mapped onto the workstation viewport in its entirety. To ensure equal scaling on each axis, the workstation transformation maps the workstation window onto the largest rectangle that fits within the workstation viewport such that:

d) aspect ratio is preserved;

e) the lower left-hand corner of the workstation window is mapped to the lower left-hand corner of the workstation viewport.

Thus, space is left unused at the top or right side of the workstation viewport if the aspect ratios of the workstation window and workstation viewport are different.

When a display is addressed through a window manager, the window provided by the window manager is analogous to the display surface of a conventional display. The limits of the display space can be inquired from the workstation description table. The workstation viewport is then defined relative to the display space in the usual way. The window manager is required to retain the contents of the display surface and do damage repair correctly.

All workstation transformations are set by default to map LDC space [0,1] × [0,1] onto the whole of the workstation display space. If the display space is not square, the same rules as above apply to achieve equal scaling on each axis.

Workstation transformations are changed by the SET WORKSTATION WINDOW AND VIEWPORT function.

7.7 Output primitives

Each output primitive has four classes of attributes (identification, source, NDC and logical). When a primitive is created, the current values of all the attributes of the primitive in the GKS state list are bound to the primitive. The values of attributes of primitives in the NDC picture can be changed by editing (see 6.5.2).

Each workstation has a set of *bundle tables,* one per primitive class. A *bundle index* is defined for each class of bundle table. Each bundle table entry contains values for a set of logical attributes. For example, the curve bundle table has a set of entries each of which contains values for CURVETYPE, CURVEWIDTH SCALE FACTOR and CURVE COLOUR SPECIFIER accessed via the curve bundle table index.

Values for each of the logical attributes are also present in the GKS state list. The GKS state list also contains entries for an index value for each bundle table and sets of *attribute source flags* (ASFs), one for each logical attribute and grouped by primitive class. The initial values of all the ASFs are INDIVIDUAL. The bundle index and set of ASFs for a particular primitive class constitute the source attributes of that primitive class.

At the workstation, values of logical attributes of output primitives are determined in the following way:

a) for all logical attributes for which the ASF is INDIVIDUAL, the values already bound to the primitive are used directly;

b) for all logical attributes for which the ASF is BUNDLED, the values bound to the primitive are replaced by the values of the corresponding attributes in the bundle table entry indexed by the bundle index bound to the primitive.

Figure 7 shows diagrammatically how the binding takes place. The logical attributes for a particular primitive are bound to a primitive on creation from the GKS state list. At the same time, the attribute source flags (represented here by arrows, up representing INDIVIDUAL, down representing BUNDLED) together with the

AVA Documents Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

class index into the bundle table are also bound. All these attributes are available for editing in the NDC picture. In the creation of the logical picture, the index into the bundle table selects a bundle table entry. The direction of the arrow decides whether the INDIVIDUAL values already bound to the primitive are retained, or the values in the bundle table entry are bound to the primitive.

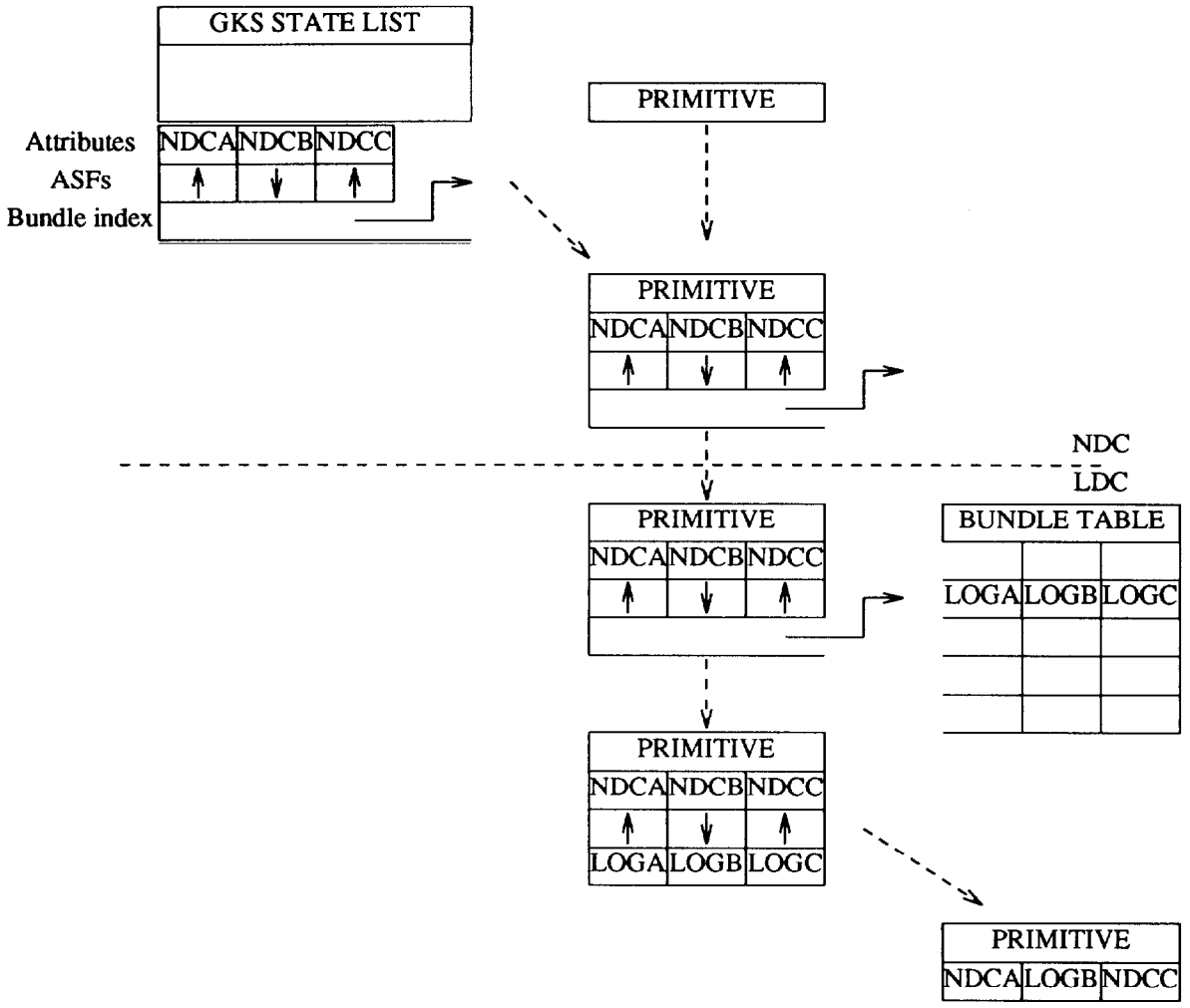

Figure 7 - Effect of different ASF settings

7.8 Colour

Colour is specified in a number of different situations. It may be an attribute of a primitive, in which case it is specified by a colour specifier. It may be part of a pattern for area primitives, or part of the primitive itself (namely for CELL ARRAY); in both these cases, colour is specified by a colour array specifier.

Colour specifiers may be of two colour types. If the colour type is indirect the colour specifier contains a *colour index*, an index into the workstation *colour table*. Alternatively the colour specifier contains a colour model name together with a specification of colour coordinates in the colour space of that model. Colours in the workstation colour table are defined by a colour model name and colour coordinates in that model.

A *colour array specifier* specifies an array of colours. This may be either an array of colour indices or a colour model name and an array of colour coordinates in that colour model.

Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

Colours are associated with a colour index by the function SET REPRESENTATION (see 7.9). The function INQUIRE WORKSTATION STATE LIST ENTRY can be used to determine the colour value associated with a colour index. The parameters of these functions describe colour as a specification of colour coordinates in the colour space of the specified colour model. The number of coordinates necessary to specify colour, and their interpretation, depends on the colour model.

Four colour models, RGB, CIELUV, HSV and HLS are predefined. The colour models RGB and CIELUV shall be provided. Note that some enumeration values are reserved for registration. The four predefined colour models are described in Annex E.

On each workstation, there is one colour table into which all the colour indices point. The size of the colour table is workstation dependent but entries 0 and 1 always exist. Entry 0 corresponds to the background colour. The background colour is the colour of the display space after it has been cleared. Entry 1 is the default foreground colour and entries higher than 1 correspond to alternative foreground colours. INQUIRE SPECIFIC WORKSTATION DESCRIPTION TABLE ENTRY can be used to determine the predefined colour representations for a workstation. The colour coordinates are returned in an appropriate colour model for the workstation.

When colours are realized they are mapped to the nearest available on the workstation. Some workstations do not have the capability to change the background colour, and in this case the mapping of a specific colour to the nearest available for the background colour may be different from the mapping of the same colour for the foreground colours.

The 'colour characteristics of display' entry in the workstation description table describes the colour capabilities of the display and provides information that enables the application to make best use of the colour capabilities of the workstation. For workstations capable of producing precise colours the 'colour characteristics of display' entry provides sufficient information to the application to enable precise colour specifications to be set up. The information returned is workstation dependent.

Some workstations are not capable of displaying colours (for example, workstations only capable of displaying colours with equal red, green, and blue intensities or workstations only capable of displaying colours which are different intensities of the same colour); these are referred to as monochrome workstations. Whether a workstation is capable of colour is recorded in the 'colour available' entry in the workstation description table. If the 'colour available' entry is set to MONOCHROME, the intensity is computed from the colour values in a workstation dependent way (see Travis, Annex F).

The function CONVERT COLOUR can be used to convert colour specifiers to colour coordinates in a specified colour model. The conversion is performed according to the conversion algorithms used by a specified workstation and hence can be used to determine how a particular colour specifier will be realized by a particular workstation.

7.9 Setting representations

The function SET REPRESENTATION defines an entry in a bundle table, the pattern table or the colour table of a workstation. Some standard definitions for data entries are contained in the workstation description table and are used as initial values. The application program can select a standard definition or can define the value of a specific entry explicitly. Only the most commonly used (or anticipated) combinations of values need be predefined for each workstation. At least these predefined entries with indices up to the minimum number of predefined entries are distinguishable from each other. Other combinations of values can be set by the SET REPRESENTATION function, possibly after inquiring the workstation capabilities. The tables which are on every workstation are:

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

curve bundle table
marker bundle table
area bundle table
pattern table
character bundle table

colour table

The values in these tables may be changed at any time. Changes only effect the rendering of primitives in the logical picture. If the change of representation were to have any effect on the backdrop, the backdrop is removed from this workstation. It is possible to set an entry to undefined as a means of recovering storage.

7.10 Removing a backdrop

The backdrop on a workstation can be removed by invoking the function REMOVE BACKDROP. (The backdrop can also be removed by invoking the function CLEAR WORKSTATION.) Changes to the workstation state list can change the NDC picture. If such changes occurred while similar output was routed to the backdrop, the backdrop is removed since it is not always possible to update the backdrop.

7.11 Visual effect state

The function SET WORKSTATION VISUAL EFFECTS allows the application to suppress and resume updating of the logical and realized pictures. This allows a group of updates to be made to appear atomic.

When visual effects state is SUPPRESS, changes have no effect on the logical and realized pictures. When routed to the backdrop, any primitive should be routed to the open workstations irrespective of the workstations' visual effects state. When visual effects state is ALLOW, updating of the logical and realized pictures is continuous. When the visual effects state is changed from SUPPRESS to ALLOW, changes made whilst the visual effects state was SUPPRESS now produce effects in the logical and realized pictures.

A function GET WORKSTATION STATUS allows the application to ascertain when all the suppressed effects (for example, from a group of updates) have been accomplished.

7.12 Realized metafile

The contents of the realized picture may be captured and stored for future use or transmission to another system. The function COPY REALIZED PICTURE TO REALIZED METAFILE will store the realized picture as a picture on the specified metafile. Blank pictures may be inserted in the realized metafile. This can be achieved by invoking the function COPY BLANK REALIZED PICTURE TO REALIZED METAFILE. The picture can be recovered at a later time and added to the current backdrop by invoking COPY REALIZED METAFILE PICTURE TO BACKDROP. The backdrop cannot be captured in a realized metafile. The external representation of the realized metafile is not defined in this International Standard.

7.13 Logical input devices

7.13.1 Introduction

A workstation may have one or more logical input devices for each input class. A logical input device is identified by a logical input device identifier which consists of a workstation identifier, a device class and the device number within that class on that particular workstation.

The application program has some degree of control over logical input devices in that it can supply an initial value for the device, set an area of the display space for the display of prompt and echos and select prompting and echoing techniques to be used.

The application program can also define the composition of compound logical input devices in terms of the *measures* and *triggers* provided by the workstation. A measure is a value which is determined by one or more physical input devices and a mapping from the values delivered by the physical devices. A trigger is a

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

condition which determines when a logical input value is delivered by a logical input device.

7.13.2 Initialization of logical input devices

Logical input devices may only be initialized when in REQUEST mode. The INITIALIZE LOGICAL INPUT DEVICE function provides the following information to a device via the workstation state list (if the INITIALIZE LOGICAL INPUT DEVICE function is not invoked, then default values apply):

a) An initial value appropriate to the device. If the initial value violates the rules, an error occurs and the workstation state list is unchanged.

b) A prompt and echo type that selects the prompting technique and, if echoing is on, the echoing technique for a logical input device. An implementation dependent prompt and echo type (type 1) is required for all logical input devices. Further prompt and echo types are defined but not required. Prompt and echo types above those are reserved for registration (see 4.2). Prompt and echo types less than 0 are device dependent.

c) An echo switch that controls whether echoing is on or off.

d) An echo area (xmin,xmax,ymin,ymax) in device coordinates. Input device implementations may use the echo area for certain prompt and echo types to display prompts or echoes. The specified echo area need not be within the current viewport. An input device implementation is not required to use the specified echo area; in a windowing environment echoing may be provided outside the display space.

e) A data record. Some input classes have mandatory control values in the data record. Some prompt and echo types within an input class also have mandatory control values in the data record. These values occupy well defined places in the data record. In any data record used in initializing an input device, values mandatory for the input class, if any, appear first followed by values mandatory to the prompt and echo type, if any. Depending on the device and prompt and echo type, the data record may contain other (additional) information.

When a logical input device is REQUESTed, or when it is set to EVENT or SAMPLE mode, its measure is set to the initial value from the workstation state list, unless this is not a valid measure for the device. If it is not a valid measure for the device, the measure is set to a device dependent value, except for PICK devices, for which the measure is set to NOPICK.

Prompt and echo types describe both the prompt, which informs the operator that the device is available, and the echo, which informs the operator of the state of the measure. In addition, an implementation dependent acknowledgement of successful trigger firings is provided.

7.13.3 Definition of logical and composite input devices

The application program can construct logical input devices of composite types using the function DEFINE LOGICAL INPUT DEVICE. Each workstation supporting input devices provides a number of measures and triggers that map onto physical devices associated with the workstation. The application can specify which of these measures and triggers are to be used in constructing the measure and trigger processes for a specified device. The value returned by such a device is a sequence of values of basic or constructed types. Satisfaction of any of the trigger conditions specified causes the corresponding logical input device's trigger to fire.

For example, an input device could be defined which returns two measures from a single physical device such as a mouse. The first measure could be derived from the mouse's position and the second measure could be derived from the current positions of the buttons on the mouse.

7.14 Sending messages to a workstation

The MESSAGE function allows a character string to be sent to a workstation. The application program has no control over the position and appearance of the character string and an implementation is allowed to place the string on a device distinct from, but associated with, the workstation. For example, in a windowing environment the message could appear in a pop-up window.

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

8 Output primitives

8.1 Introduction

Clause 6 contains a description of the overall capabilities of the output primitive classes and their attributes. This clause gives details of the output primitives and attributes in each class. The logical attributes for each class are given in the following table. The individual subclauses give their meaning.

Output primitive class	Logical attributes
CURVE	CURVETYPE CURVEWIDTH SCALE FACTOR CURVE COLOUR SPECIFIER
MARKER	MARKERTYPE MARKERSIZE SCALE FACTOR MARKER COLOUR SPECIFIER
AREA	INTERIOR STYLE INTERIOR STYLE INDEX INTERIOR COLOUR SPECIFIER EDGE FLAG EDGETYPE EDGEWIDTH SCALE FACTOR EDGE COLOUR SPECIFIER
CHARACTER	CHARACTER FONT AND PRECISION CHARACTER EXPANSION FACTOR CHARACTER SPACING CHARACTER COLOUR SPECIFIER
IMAGE	
DESIGN	

8.2 Curve output primitives

8.2.1 Curve primitives

The output primitives in the class curve are:

a) SET OF POLYLINE: generates a set of curves each of which is a sequence of connected lines defined by a point sequence.

b) SET OF NURB: generates a set of curves each of which is a nurb.

c) SET OF CONIC SECTION: generates a set of curves each of which is a conic section.

A nurb path is defined as follows. The nurb is specified by its order k (degree $k-1$), the knot sequence $(T_1, \cdots, \cdots, T_{n+k})$, where n is the number of control points, and the set of control points $\{\mathbf{P}_i\}^n$, a set of weights $\{W_i\}^n$, a minimum $t_{\min}$ and maximum $t_{\max}$ of an independent parameter t. For non-rational control points, the weights $\{W_i\}$ are all taken as 1. The knot sequence must satisfy (simultaneously):

$$T_k \le t_{\min} \le t \le t_{\max} \le T_{n+1}$$

$$T_1 \le T_2 \cdots \le T_{n+k-1} \le T_{n+k}$$

The curve is the (infinite) set of points **P** for all possible values of t between $t_{\min}$ and $t_{\max}$.

Documents
Document Sharing Hub
Full document details: ISO IEC 07942-1-1994

$$\{\mathbf{P}(t) : t_{\min} \le t \le t_{\max} \text{ and } \mathbf{P}(t) = \frac{\sum_{i=1}^{n} \mathbf{P}_i W_i B_{i,k}(t)}{\sum_{i=1}^{n} W_i B_{i,k}(t)}\}$$

where the Basis Functions $B_{i,k}$ are defined (using the knot sequence) as:

$$B_{i,0}(t) = \begin{cases} 1 & \text{if } T_i \le t < T_{i+1} \text{ and } T_i < T_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$B_{i,p}(t) = \frac{(t-T_i)B_{i,p-1}(t)}{(T_{i+p-1}-T_i)} + \frac{(T_{i+p}-t)B_{i+1,p-1}(t)}{(T_{i+p}-T_{i+1})}$$

where $p=1..k$. In the case of multiple identical knot values some denominators evaluate to 0, in such cases, as part of the above definition the indeterminate quantity $\frac{0}{0}$ is considered to be 0.

For the SET OF CONIC SECTION primitive, each conic section is defined by a 3 × 3 matrix, A, which specifies the conic and two points which define the start and end points of the conic section. The equation of the conic is:

$$\mathbf{X}^T A\, \mathbf{X} = 0 \quad \textit{where} \quad \mathbf{X} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

The last parameter of the conic section definition is a sense flag which indicates whether it is the clockwise or anti-clockwise section that is required when the conic is a circle or an ellipse. A number of utility functions EVALUATE CIRCULAR ARC 3 POINT, EVALUATE CIRCULAR ARC CENTRE, EVALUATE ELLIPTIC ARC, EVALUATE HYPERBOLIC ARC, and EVALUATE PARABOLIC ARC are provided (see 12.14) to define the conic section for circular, elliptic, hyperbolic and parabolic arcs. For EVALUATE CIRCULAR ARC CENTRE, EVALUATE ELLIPTIC ARC, and EVALUATE HYPERBOLIC ARC the start and end points of the arc are defined as the intersection of the conic with lines from the centre in the directions of specified start and end vectors.

8.2.2 Curve attributes

The curve logical attributes are CURVETYPE, CURVEWIDTH SCALE FACTOR and CURVE COLOUR SPECIFIER.

The defined curvetypes 1 to 5 are solid, dashed, dotted, dashed-dotted, and dashed-dotted-dotted. The curvetype specifies a sequence of line segments and gaps which are repeated to define the rendering of the curve primitive. It is workstation dependent whether this sequence is restarted or continued at the start of each curve in a set and at the start of a scissored piece of a curve primitive. For curve primitives defined by vertices, the sequence shall be continuous across vertices.

The curvewidth is calculated as a nominal curvewidth multiplied by the curvewidth scale factor. This value is mapped by the workstation to the nearest available curvewidth in the realized picture.

8.3 Marker output primitives

There is a single output primitive in the class marker which is POLYMARKER. This generates a set of symbols of one type centred at given positions.

The marker logical attributes are MARKERTYPE, MARKERSIZE SCALE FACTOR, and MARKER COLOUR SPECIFIER.

Markertypes 1 to 5 are dot, plus sign, asterisk, circle and diagonal cross each centred on the positions they are identifying.

The marker size is calculated as a nominal size multiplied by the markersize scale factor. This size is mapped by the workstation to the nearest available marker size in the realized picture. Marker type 1 is always

AVA Documents Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

displayed as the smallest displayable dot in the realized picture irrespective of the markersize scale factor.

The marker is visible if, and only if, the marker position is not scissored. The scissoring of partially visible markers is workstation dependent.

8.4 Area output primitives

8.4.1 Area primitives

The output primitives in the class area are:

a) SET OF FILL AREA: generates a set of areas each of which is defined by a closed sequence of connected points.

b) SET OF CLOSED NURB: generates a set of areas each defined by a nurb which is closed by the first point being connected to the last.

c) SET OF ELLIPTIC SECTOR: generates a set of areas each defined by an elliptic arc. Each area is closed by lines from the ends of the arc to the centre of the ellipse.

d) SET OF ELLIPTIC SEGMENT: generates a set of areas each defined by an elliptic arc. Each area is closed by a line joining the ends of the arc.

e) SET OF ELLIPTIC DISC: generates a set of areas each defined by a complete ellipse.

The elliptic arcs in the SET OF ELLIPTIC SECTOR and SET OF ELLIPTIC SEGMENT primitives are defined by a 3×3 matrix which defines the ellipse, in the same way as the SET OF CONIC SECTION curve output primitives (see 8.2), and two points which specify the start and end point of the elliptic arc. Finally, a sense flag defines whether it is the clockwise or anti-clockwise arc from the start point to the end point. The utility functions EVALUATE CIRCULAR ARC 3 POINT, EVALUATE CIRCULAR ARC CENTRE and EVALUATE ELLIPTIC ARC are provided to define the elliptic arc.

The SET OF ELLIPTIC DISC primitive is defined by a 3×3 matrix which defines the ellipse. The utility functions EVALUATE CIRCLE and EVALUATE ELLIPSE are provided to define the 3×3 matrix.

Area primitives allow the specification of areas with holes or disjoint regions that should be treated as a single entity. For each area primitive, the areas may be hollow or filled with a uniform colour, a pattern, or a hatch style, with or without edges (the boundary of each closed region).

8.4.2 Definition of interior

The set of interior points of area primitives is defined in the following way (see figure 8). For any point not on the boundary of the area primitive, if the number of intersections between the boundary of the area primitive and any ray from the point to infinity (not passing through a vertex) is odd, the point is interior, otherwise it is outside. If a point is interior, it is included in the area to be filled, subject to scissoring. When an area primitive is scissored, the resulting new boundaries become part of the area boundaries (see figure 9). These boundaries are not displayed as edges. Multiple subareas may be generated.

8.4.3 Area attributes

The area primitives have the NDC attributes PATTERN REFERENCE POINT and PATTERN SIZE. If the PATTERN SIZE is (SX,SY), this defines the pattern height vector as (0,SY) and pattern width vector as (SX,0). These values are defined in WC and are subject to the same transformations as the geometric data contained in the definition of the primitive. The usage of these area NDC attributes is described later in this subclause.

The logical attributes of area primitives are INTERIOR STYLE, INTERIOR STYLE INDEX, INTERIOR COLOUR SPECIFIER, EDGE FLAG, EDGETYPE, EDGEWIDTH SCALE FACTOR and EDGE COLOUR SPECIFIER.

AVA Documents Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

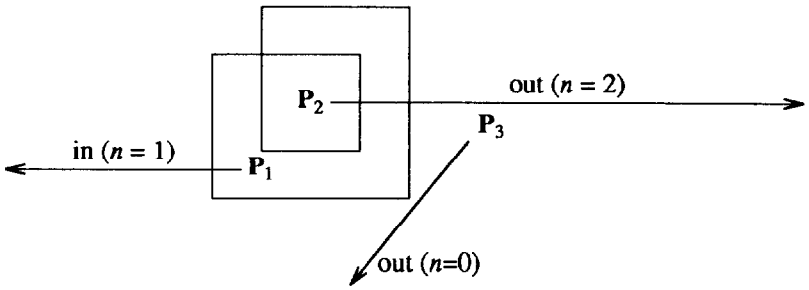

Figure 8 - The inside of an area

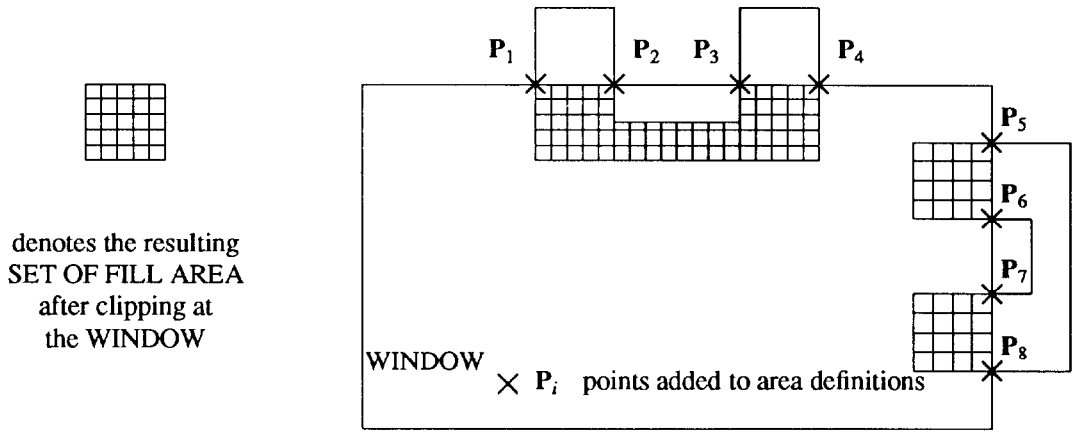

Figure 9 - Example of SET OF FILL AREA scissoring

The INTERIOR STYLE is used to determine in what style the areas should be filled. It has the following values and meanings:

a) HOLLOW: No filling, but draw the bounding curve of the area, using the INTERIOR COLOUR SPECIFIER currently selected. The curvetype and curvewidth are implementation dependent. The bounding curve is drawn along boundaries created by scissoring.

b) SOLID: Fill the interior of the area using the INTERIOR COLOUR SPECIFIER currently selected.

c) PATTERN: Fill the interior of the area using the INTERIOR STYLE INDEX currently selected as an index into the pattern table. In this context, the INTERIOR STYLE INDEX is sometimes referred to as the pattern index.

d) HATCH: Fill the interior of the area using the INTERIOR COLOUR SPECIFIER and the INTERIOR STYLE INDEX currently selected. The INTERIOR STYLE INDEX is used as a pointer into the list of hatch styles, in which case it is sometimes referred to as the hatch index.

e) EMPTY: No filling.

For interior style PATTERN, the pattern index selects among different patterns. The position in the pattern table defines a pattern representation, which specifies a colour array specifier of dimension (DX × DY). The size and position of the start of the pattern are determined by a pattern box. The pattern box, which is a parallelogram in NDC, is defined by the pattern width vector and the pattern height vector located relative to the PATTERN REFERENCE POINT. The pattern box is conceptually divided into a grid of DX × DY cells of

Documents Document Sharing Hub

Full document details: ISO IEC 07942-1-1994

equal size. The colour array is associated with the cells as follows: the element (1,DY) is associated with the cell having the PATTERN REFERENCE POINT at one corner; elements with increasing first dimension are associated with successive cells in the direction of the pattern width vector; elements with decreasing second dimension are associated with successive cells in the direction of the pattern height vector. The attributes defining the pattern box are subject to all the transformations producing a transformed pattern box. The pattern is mapped onto the area by conceptually replicating the transformed pattern box in directions parallel to its sides until the interior of the complete area is covered.

Mapping the transformed pattern cells to the pixels of a raster display is performed by the following rules:

f) if the location of a pixel lies inside the parallelogram defined by the transformed cell, its colour is set;

g) the pixel is assigned the colour of the cell corresponding to the pixel's centre.

For a workstation which can implement patterns but not transformable patterns, a suitable action is to generate non-transformed patterns to fill an area.

For interior style HATCH, the hatch index selects among hatch styles. The defined hatch styles 1 to 6 are horizontal equally spaced parallel lines, vertical equally spaced parallel lines, positive slope equally spaced parallel lines, negative slope equally spaced parallel lines, horizontal/vertical cross hatch and positive slope/negative slope cross hatch. The ideal angle for positive slope hatch style is +45° and the ideal slope for negative slope hatch style is +135°.

Whether hatching is affected by transformations or not is workstation dependent.

Interior styles HOLLOW, SOLID and EMPTY are available on every workstation. It is workstation dependent which of the interior styles PATTERN and HATCH are available.

The other logical attributes permit independent control of the edges of area primitives. They are EDGE FLAG, EDGETYPE, EDGEWIDTH SCALE FACTOR and EDGE COLOUR SPECIFIER.

The EDGE FLAG may have values ON and OFF. When the value is OFF, the edge is not drawn. When the value is ON, the edge is rendered according to the other edge attributes. Edges are drawn on top of the interior. Conceptually edges and interior are disjoint; in practice, they can overlap. Parts of edges that are scissored are not visible. The defined edgetypes are the same as curvetypes and have the same meanings. The edgewidth is calculated as a nominal edgewidth multiplied by the edgewidth scale factor. This value is mapped by the workstation to the nearest available edgewidth in the realized picture. For example, if the value is less than or equal to zero, the thinnest available edgewidth is used.

For an area primitive with interior style HOLLOW, the rendering of the bounding curve may be overlaid by the rendering of the edge since the boundary is considered part of the interior. The bounding curve may be visible between gaps of non-solid edgetypes, or around the rendering of the edge.

New boundaries created by scissoring are not displayed as edges.

Area primitives with interior style EMPTY are visible only if edge flag is ON.

8.5 Character output primitives

8.5.1 Introduction

The only output primitive in the class character is TEXT which generates a sequence of glyphs specified as a sequence of character codes relative to a specified text position.

The NDC attributes associated with the TEXT primitive are TEXT HEIGHT, TEXT UP VECTOR, TEXT PATH, TEXT SKEW ANGLE, and TEXT ALIGNMENT. It also has the logical attributes of the character class associated with it, namely CHARACTER FONT AND PRECISION, CHARACTER EXPANSION FACTOR, CHARACTER SPACING and CHARACTER COLOUR SPECIFIER. The attribute CHARACTER FONT AND PRECISION defines font index and precision.

The definition of the graphical output proceeds as follows:

AVA Documents Document Sharing Hub

Full document details: ISO IEC 07942-1-1994