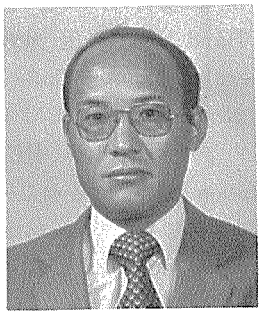
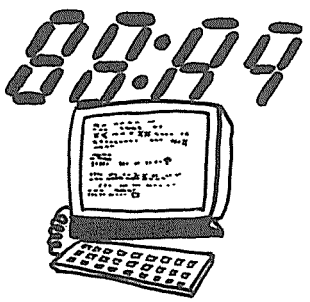


吳 吉 祿
韓國電子通信研究所
컴퓨터 연구부장/工博

Graphics Standards 연구



1. 서론

Computer graphics는 한번에 보다 많은 정보를 사용자에게 전달할 수 있는 매체로서, 나날이 진보되고 있는 Computer의 처리능력을 충분히 활용할 수 있는 Man-machine interface로서 새롭게 부각되고 있다. 이러한 Man-machine interface로서의 graphics는 새로운 VLSI graphics display controller의 개발, 더욱 더 싸지고 있는 대용량 RAM의 개발, Raster graphics technology의 활용 등의 기술적인 성과와, Engineering workstation, CAD/CAM, Office automation 등 새로운 응용분야의 개척등으로 그 활용 범위를 넓혀가고 있다.

그러나 Computer graphics hardware는 device dependent한 software를 사용함으로 인하여 graphics program을 다른 system에서 사용할 수 없는 문제점을 노출하여 왔다. 이러한 software portability 문제를 해결하기 위하여 graphics software 표준화에 관한 연구가 활발하게 진행되어 왔다. 이러한 graphics standards로, 현재 ACM의 Core system과 ISO에 제출된 DIN의 GKS (Graphical Kernel System) 등이 있다.

일반적으로, 기본적인 graphical function을

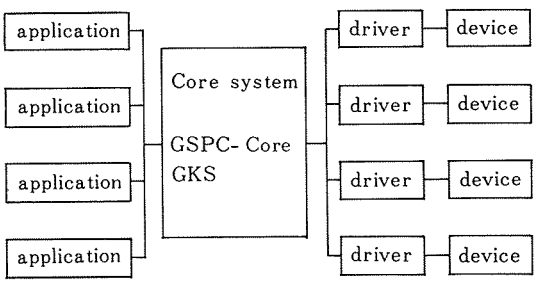


그림 1. Graphics system with application interface and device interface

실현하는 기본 system을 「basic Core」라고 부르며, 기본 기능뿐만 아니라 확장된 복잡한 기능들을 제공하는 system을 「rich Core」라고부른다.

2. 표준화의 필요성

graphics system에서 왜 표준화가 필요한가는 다음과 같이 요약될 수 있다.

- Device independence
- Program portability (computer independence)
- Programmer portability

Device independence는 한 응용 program을 다른 여러가지 display devices에서 사용할 수 있게 한다. 응용 programmer는 target device의 특성에 구애됨이 없이 virtual space에 graphics primitive function을 이용하여 나타내고자 하는 그림을 표현할 수 있으며, graphics support package는 응용 program의 graphics 요구조건을 target device의 graphics vocabulary로 mapping한다. 즉, virtual device interface로 각각의 display device를 support한다. 또한 새로운 graphics device를 추가시켜 사용할 경우에도 새로운 graphics device의 driver routine만을 추가시켜서 기존의 program을 활용할 수 있다.

Program portability는 표준화의 가장 큰 이점이다. 하나의 graphics 응용program이 작성되었을 경우, 이를 다른 computer에서도 사용할 수 있다.

예를 들어, VAX computer에서 사용하던 표준화된 graphics 응용program은 Apollo computer에서도 사용할 수 있다. graphics가 표준화되기 이전까지는, 이러한 문제는고급언어(예를 들면, Fortran)를 사용하여 이를 해결하여 왔다. 이러한 Source program portability는 새로운 Graphics application program package 개발에 드는 노력과 시간을, 이미 만들어진 기존의 program을 활용하여, 경감시킬 수 있다.

Programmer가 한 graphics package를 사용하다 다른 것을 사용하게 될 경우, 새로운 이질적 개념 및 programming 기법을 배워야만 한다.

graphics programming은 일반적인 언어 programming과 비교시, 용어나 개념상의 유사성이 공존하지 않으므로, graphics 응용 programmer의 재교육이 필요하다.

따라서, 표준화된 graphics 소프트웨어를 사용함으로써 재교육의 필요성을 경제적, 시간적으로 감소시킬 수 있다.

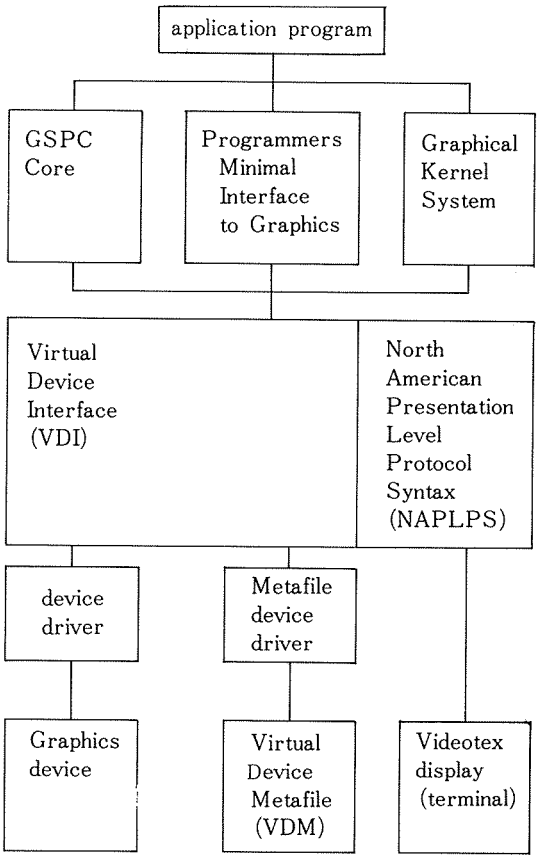


그림 2. **Computer Graphics system Conceptual Model**

3. 표준화의 과정

1970년대 초기에 있어서, graphics 표준화 연구의 목표는 Pen plotter, Direct view storage tube, Refresh calligraphic display, Raster scan display 등의 여러 종류의 device에 출력시킬 수 있는 그림의 묘사를 정의하는 것이었다.

ACM의 SIGGRAPH (Special Interest Group on Computer Graphics)는 1974년에 「Machine Independent Graphics」에 관한 workshop을 열

었는데, 이는 공식적인 graphics 표준 제정의 이정표가 되었다.

1976년 5월 프랑스 Selliac에서 SIGGRAPH GSPC(Graphics Standards Planning Committee)가 「Graphics Standards Methodology」를 주제로 workshop을 개최하였으며, 여기서 두가지 task가 설정되었다. 첫째는 program이식을 위한 방법을 개발하는 것이며, 둘째는 Core system을 위한 기능적인 사양을 정의하는 것이었다.

1977년, GSPC는 입력 및 출력 device capabilities의 범위를 포함한 graphics 표준서를 처음으로 만들었다. 1979년, SIGGRAPH conference에서 1977년 사양을 좀더 보완하여, Core system의 방법과 기능적인 사양에 관한 graphics 표준 proposal이 개정되었다. 이 제안서에는 Core system의 Raster graphics 확장, Metafile (Device independent plot file), Distributed graphics system에 관한 기능적인 사양을 포함하고 있다.

GSPC의 연구는 ANSI Technical Committee X3H3, Computer Graphics programming 언어 구성의 출발점이 되었다.

이 위원회 X3H3는 다음과 같이 구성되어 있다.

3.1. X3H31 : Full Graphics Language Standard
graphics primitives, attributes, viewing operations, picture segmentation, input function 및 2-D, 3-D의 일반적 control을 포함.

3.2. X3H32 : Reference Models
graphics software system의 논리적 정의 및 segmentation

3.3. X3H33 : VDI, VDM
Virtual Device Interface (VDI)
device independent software, device dependent driver사이의 data 교환 및 control의 functional syntatic specification
Virtual Device Metafile (VDM)
graphics data 및 control을 전달하고 저장하는 mechanism

3.4. X3H34 : Binding
graphics language implementation을 포함

3.5. X3H35 : Programmer's Minimal Interface to Graphics (PMIG)
2-D 동작에 제한한 Full language standard의 subset포함

이러한 ANSI 위원회는 U.S. 표준인 Core system을 개발하고 있으며, 유럽 graphics 표준은 서독에 의해 주로 연구되고 있는 GKS를 개발하고 있다.

GKS는 Core system으로부터 많은 영향을 받았으며 기능, 개념, 방법상에서 유사점이 많다. 또한, GKS는 ISO에 의해 공식적인 국제 2D graphics 표준으로 설정될 것이다.

X3H3 위원회는 GKS에 대하여 Core system과 호환성이 있게 만들 것을 종용하고 있으며, GKS를 2D 표준으로 받아들일 전망이다.

graphics 표준화는 많은 기업과 국가의 이해가 관련되어 있으므로 궁극적으로는 GKS가 2D graphics 표준이 되리라 예측된다. 이러한 ANSI 및 ISO의 노력에 더하여, 통신분야에 있어서 graphics 표준이 1981년 중반경 나타났다.

AT&T는 Teletex나 Videotex와 같은 통신

표 1. Graphics standards의 표준화 과정

1974	ACM SIGGRAPH GSPC 구성
1976	프랑스 Selliac에서 컴퓨터 그래픽스의 방법에 관한 국제 회의 열림
1977	GSPC Core system 및 DIN Graphical Kernel System specification 발간됨
1979	GSPC Core system 및 GKS규격 개정됨 GKS가 ISO working draft로 채택됨 Virtual Device Interface (VDI) 및 Virtual Device Metafile (VDM) 연구 시작 Canada가 Telidon (Videotex) 시도 인정
1981	AT&T에 의해 NAPLPS (North American Presentation Level Protocol Syntax) 발표
1982	GKS가 Proposed ISO standard로 등록 VDI, VDM, NAPLPS가 industry support 받음
1984	GKS가 공식 ISO standard가 될 전망 NAPLPS, VDI, VDM이 U.S. Standards로 채택될 전망

system간의 문서 및 화상의 전달을 위한 화상묘사기법인 PLP(Presentation Level Protocol)을 발표했다. AT&T의 Protocol은 Canada의 Telidon system의 화상묘사 명령을 사용하였다.

이 명령은 ANSI graphics 표준의 VDI(Virtual Device Interface) 출력 primitive와 유사하나, 현재 같은 syntax는 아니다. AT&T의 연구와 Canada의 Telidon은 NAPLPS(North American Presentation Level Protocol Syntax)로 합쳐질 것이며, 이는 특히 마이크로컴퓨터 그래픽스에 영향을 줄 것이다.

4. Core system의 분석

Computer와 Graphics devices를 사용하여 기본적인 graphical capabilities를 실현할 수 있는 Computer graphics system은 「Core System」이라 정의되고 Core system의 기능을 사용하여 graphical objects를 나타내는 system을 「Modelling System」이라 한다. 예를 들어, 여러가지 기계 부품의 geometrical models을 정의하고, 계산하고, 저장하고, 표시하는 system은 geometric modelling system이고, 이러한 models의 graphical presentation 및 사용자의 interaction은 Core system의 task가 된다.

Core system은 다음과 같은 properties를 지닌다.

- independence of a specific computer
- independence of a specific devices
- independence of a specific programming language
- independence of a specific application area

Core system은 일반적으로 인정된 reference model에 기초를 두어야 하며, 또한 그러한 Core system의 표준화에 의한 개발이 바람직하다. 이는 Computer graphic programs의 program portability를 증진시킬 수 있을 뿐만 아니라 Computer graphics education의 uniform base를 제공할 수 있다.

표준화된 Computer graphics system은 다음과 같은 분야를 정의하고 있다.

- a standard functional interface for all kinds of application
- a standard device interface to all kinds of graphics devices
- a standard interface for storage and transfer of graphical information (graphics metafiles)

이러한 표준화된 Core system을 실현하기 위해서 GSPC-Core proposal이 제안되었으며, 이는 3D graphical Core system을 위한 것으로 ACM-SIGGRAPH의 Graphics Standards Planning Committee(GSPC)에 의해서 고안되었다.

1977년에 proposal이 발간되었으며 1979년에 기능을 확장하여 proposal이 개정되었다. 1979년부터 GSPC proposal은 ANSI committee X3H3에 의해서 comprehensive한 3-D graphics 표준으로 만들기 위한 작업이 진행중이다.

대부분 미국내에서 상업화되어 사용되는 graphics package는 GSPC-Core 1979년 version의 사양에 따른 것들이 널리 활용되고 있다. Graphics 표준인 Core system은 크게 다음과 같은 6가지의 기능들로 나누어진다.

4.1 Output Primitives 및 attributes

화면에 나타날 objects는 2D 또는 3D graphical world로 표시된다. 이 object는 world coordinate system 상에서 output primitive 기능을 호출함으로써 나타내어진다. output primitive 기능을 호출한 결과,

- creation of output primitive
- the parameters to the function (coordinates, text string or marker type)
- viewing transformation
- the current value of the primitive attributes가 정해진다.

Output pimitive는 LINE, POLYLINE, TEXT, MARKER, POLYMARKER, POLYGON 등 6가지가 있으며 current position 및 world coordinates에 의해서 동작된다. output primitive의 attributes의 종류는 line style, line width, intensity, color, character font, character size, character spacing, character quality, character plane, pick identifier 등이다.

4.2 Viewing Transformation

Viewing transformation은 나타날 graphical world의 특정지역을 선택하며, 선택된 지역의 objects가 어떻게 view space로 mapping될 것인가를 지정한다.

View space는 사용되는 device의 2D output surface의 rectangular area이다. view space상에 위치를 명시하는 좌표계는 normalized device coordinate system (NDC)이라 불린다.Core system의 viewing capabilities는 2D 및 3D의 subset이다. 2-D의 경우, viewing transformation은 world coordinates의 window와 view space의 viewport에 의해서 지정된다.

Window는 X-Y world coordinate plane상의 어떠한 rectangle도 될 수 있으며, viewport는 NDC상에 지정된 rectangle이다. window는 object를 clipping하기 위해 사용된다. clipping한 다음, 이 부분은 viewport에 의해 bounded된 view surface로 mapping된다.

3-D의 경우, view surface로의 clipping과 mapping에 더하여 3-D에서 2-D로의 projection을 포함하는 viewing transformation이 수행되므로 더욱 복잡하다. projection은 graphical world (또는 view plane)내의 projection plane에 의해 specified되거나, 또는 perspective projection의 경우 center of projection (eye point), parallel projection의 경우 projection direction에 의해 specified된다. 3-D viewing에는 2가지 종류의 clipping이 있다.

첫째는 depth clipping으로, projection plane에 parallel하게 두 plane 사이 (hither와 you plane)의 region으로 object를 clip한다.

둘째는, projection plane에 specified된 window내에서 projection이 있게끔 object를 clip한다. 이 두가지 clipping rules은 3-D상에 visible region 또는 view volume을 define한다.

perspective projection의 경우 view volume은 truncated rectangular pyramid이며, parallel projection의 경우, view volum은 finite parallel piped이다. view surface상에서 window에서 viewport로의 mapping은 2-D의 mapping과 유사하다.

4.3 Picture Segmentation과 Modification

view surface상에 display되는 picture는 하나 이상의 segment로 정의된다. application program은 picture의 parts를 변화시키기 위해서 multiple segments를 사용한다. picture의 change는 다음과 같은 경우에 일어난다.

- segment가 output primitives로부터 constructed될 때
- segment가 deleted될 때
- segment의 dynamic segment attribute가 변화될 때

Segment는 new segment를 open하고,output primitive를 create하고, segment를 clear함으로써 construct된다. segment에는 두가지 type이 create될 수 있다.

- Retained segment
- Nonretained segment

Retained segment는 image가 여러번 display될 경우 사용된다. 또한 dynamic segment attributes를 사용하여 modify할 수 있다. Segment의 primitive는 항상 record되며, segment가 delete될 때까지 save된다. Segment를 delete하게 되면, picture에서 image가 remove된다.

Nonretained segment는 한번만 display될 때 사용된다. Storage tube display 또는 hard copy device와 같이 segments나 contents가 보존될 필요가 없을 경우 사용된다. core에는 두가지 종류의 name이 사용된다. 하나는 segment, 또 하나는 primitive에 대한 name이다. Retained segment는 application program에서 사용하는 name을 가지며, segment가 delete될때라든지, segment의 attribute가 변화되어야 할 경우 사용된다. segment의 name은, segment가 create될 때 application program에 의해 specified되며, unique해야 한다. segment는, 또한 renamed될 수 있다. outputprimitive는, pick-identifier primitive attribute에 의해 specified되는 name을 가진다. segment name의 경우와는 달리, pick-identifier는 unique할 필요가 없으며, picture의 part가 변화될 수 있도록 indicate하는데 직접적으로 사용할 수 없다. retained segment의 image는 dynamic segment attribute를 사용하여 modify될 수 있다. primitive attribute의 경우와는 달리, 주어진 segment의 image에 영향을 미치는 dynamic segment

attributes는 segment가 존재할 경우 어느 때이든지 변화할 수 있다.

Dynamic segment attributes의 종류

1) visibility

image를 visible하게 하거나 invisible하게 하는 것을 control한다. open segments가 visible할 경우, image는 output primitive가 created될 때마다 변화한다.

2) highlighting

highlighted image가 non-highlighted images 사이에서 표시되게 한다. blinking을 support하는 devices에서는 blinking이 highlighting으로 쓰인다.

3) detectability

image가 pick logical input devices에 의해 detect되게 control한다.

4.4 Image transformation

segment의 image가 view surface상에서 translated, scaled, rotated되게 한다. image transformation attribute value는 existing segment에만 해당되며, view surface에 나타나는 segment의 attribute를 반복적으로 modify한다.image transformation은 2-D objects의 dragging(repetitive repositioning) 및 2-D 또는 3-D object의 tumbling(repetitive re-orientation)에 유용하다.

4.5 Logical input device

operator의 input로 logical input device로 5가지가 support된다.

1) pick : lightpen

operator가 output primitive을 선택할 수 있게 한다. segment의 name과 output primitive의 pick-identifier가 returned value이다.

2) locator : tablet, joystick

view surface상의 position을 제공한다.

3) valuator : analog control dial

scalar value를 제공한다.

4) keyboard

character를 type한다.

5) button : program function key

특별한 button이 동작되었는가를 지정한다.

각 device는 사용시에 enable되어야 한다. 각 device는 application program에 의해 on되거나 off되는 system이 정의하는 operator feedback을 가진다. logical input device는 동작 mode에 따라 두개의 형태로 나눌 수 있다.

1) event causing : pick, keyboard, button

2) sampled : locator, valuator

4.6 Control

Core system은 multiple view surface의 사용이 가능하다. view surface는 사용되기 전에 initialize되어야 하며, initialize된 후에 view surface는 dynamic하게 select되거나 deselected될 수 있다. Core system의 status, variable의 inquiry 및 device properties의 inquiry가 제공된다. application program은 pictures의 several parts에 대한 변화가 요구되는 것을 표시할 수 있다. 이는 Core system을 batch mode로 사용하게 한다.

4.7 Core system의 implementation levels

Core system은 여러 분야의 응용에 활용될 수 있어야 한다. 예를 들어, static plotting에서부터 dynamic motion 및 real time interaction에 이르기까지 여러 분야의 응용이 생각되어질 수 있다. 그러나 많은 display device는 PICKING을 위한 lightpen이나 transformation support hardware가 없는 경우가 있다. 이러한 경우 이를 software로 simulation을 하여야 한다.

응용분야에 따라, 또한 hardware support를 쉽게 하기 위하여 implementation level의 개념이 정의되었다. 3가지의 level이 upward compatible로 input, output 및 world coordinate space의 dimension의 class로 정의된다.

(1) Output levels

a) output level 1 : basic output

b) output level 2 : buffered output

c) output level 3 : dynamic output

(2) Input levels

a) input level 1 : no input

b) input level 2 : synchronous input

c) input level 3 : complete input

(3) Dimension levels

a) dimension level 1 : two dimension

b) dimension level 2 : three dimension

(4) Hidden surface levels

a) hidden surface level 1 : no hidden sur-

face removal

b) hidden surface level 2 : temporal priority

c) hidden surface level 3 : explicit priority

d) hidden surface level 4 : full hidden surface removal

Core system의 이러한 level은 동작의 유연성, implementation cost, 요구되는 system의 performance 등에 따라서 flexible하게 이용할 수 있다.

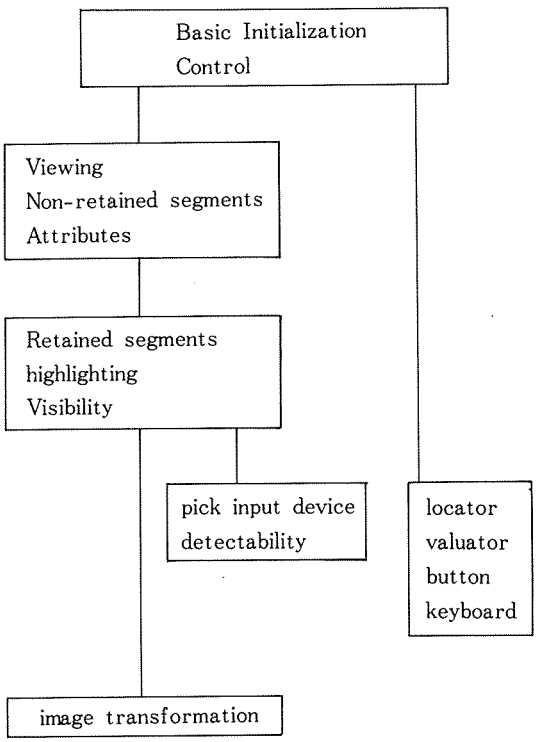


그림 3. GSPC-Core system의 implementation level 관계

5. GKS(Graphical Kernel System) 분석

2D graphical system을 위한 proposal이며 1976년부터 1979년 사이에 독일의 DIN(Deutsches Institutfur Normung)의 subcommittee NI-5.9 「Computer Graphics」에 의해서 고안되었다. ISO(International Standard Organization) TC97/SC5/WG2에 의해서 GKS 7.2version이 공식적인 국제 규격으로 채택될 전망이다.

GKS는 application program과 graphics input 또는 output 장비간에 functional interface를제공한다. 이 functional interface는 많은 graphics 장비들의 interactive 또는 non-interactive한 모든 기본적인 function을 포함한다.

picture를 처리하거나 바꾸는 일은 segment facilities, dynamic attributes, transformation 등을 통해서 이루어진다.

GKS에서는 multiple workstation의 사용이 가능하도록 되어있고, 12가지로 분류되는 GKS level은 그 function을 다 수용하지 않아도 되게 되어 있다. 다음 그림은 graphical system에서 GKS의 역할을 나타낸 것이며, 여기에서는 GKS의 특성과 그 기능을 간단히 살펴 보기로 한다.

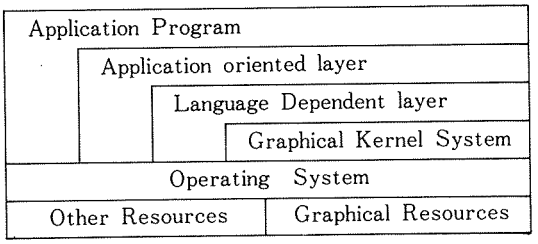


그림 4. Layer model of GKS

5.1 Output Primitives and attributes

GKS에서 행해지는 graphical output은 output primitives와 primitive attributes로 구성되어 있다. output primitive는 line drawing, character string의 printing과 같이 실제로 device에서 행해지는 기본적인 function들을 나타내며, output primitives는 linestyle, color, character height, pick identifier와 같이 device의 output primitives를 control하는 기능들을 말한다.

color와 같은 non-geometric attributes가 각각의 workstation을 효율적으로 사용할 수 있도록 하기 위해 control되어질 수 있다. GKS에서는 6가지의 output primitives를 제공하며, 각각의 Primitives에 대하여 primitive attributes가 존재한다.

- POLYLINE
- POLYMARKER
- TEXT
- FILL AREA
- CELL ARRAY
- GDP - Generalized Drawing Primitives

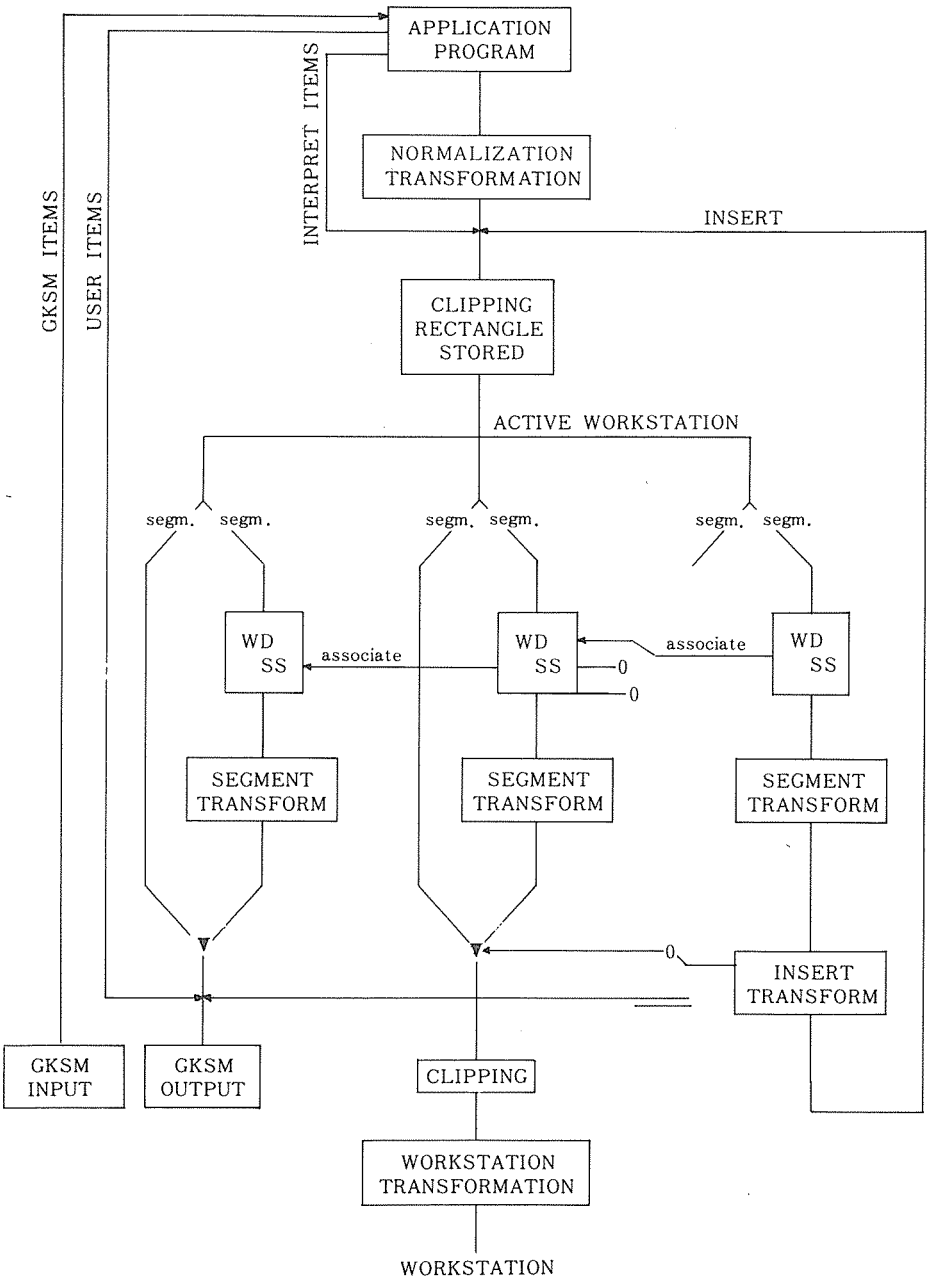


그림 5. Data flow chart for graphical output in GKS

5. 2 Workstations

GKS를 구성하는 기본 요소인 workstation은 display surface를 가지거나, 또는 keyboard, tablet, lightpen 등과 같은 input device를 가지는 하나의 unit를 나타낸다. application program은 이러한 workstation을 사용함으로써 hardware특성에 구애됨이 없이 사용할 수 있다.

이러한 workstation은 다음과 같은 특성을 가진다.

- 하나의 display surface, rectangular display space, 지정된 display area 외에 다른 display image가 없을 때, 최대 크기보다는 작은 display space의 사용이나 명시, 여러개의 linetype, text fonts, charac-

ter size 등을 제공

- 각각의 input class에 대해서 하나 이상의 input class를 가진다.
- REQUEST, SAMPLE, EVENT type의 input을 받아 들인다.
- logical input device들은 서로 관계없이 REQUEST, SAMPLE, EVENT mode로 set 되어진다.
- segement의 저장이나, 처리 또는 교환

실제 workstation이 위와 같은 모든 사항을 수행할 필요는 없으며, 각 workstation은 다음과 같은 여섯가지의 형태로 구분된다.

- OUTPUT - output
- INPUT - input
- OUTIN - output and input
- WISS - Workstation Independent Segement Storage
- MO - GKS Metafile (GKSM) output
- MI - GKSM input

5.3 Clipping and Transformations

GKS의 Clipping은 normalization transformation의 viewport와 workstation의 window 두가지로 분류할 수 있다. Clipping은 enable 또는 disable될 수 있다.

output primitives, attributes, logical input values (locators와 strokes)에 포함되는 geometric information의 transformation은 다음과 같은 세가지 coordinates system사이에 mapping을 수행한다.

1) application programmer에 의해 사용되는 world coordinates (WC)
2) 모든 workstation에 대해 uniform coordinate system을 define하기 위해 사용되는 normalized device coordinates (NDC)
3) 하나의 workstation에서 display space coordinates를 나타내는 device coordinates (DC)

output primitives와 attributes는 WC에서 NDC로 normalization transformation에 의해 행해지고, NDC에서 NDC로 segment transformation에 의해 수행되며, NDC에서 DC로 workstation transformation을 역으로 하여 수행되고, NDC에서 WC로 inverse normalization에 의해 수행된다.

5.4 Segments

segment는 manipulation이나 change를 위한 unit를 나타내며, output primitives와 primitive attributes로 구성되어질 수 있다. manipulation은 creation이나, deletion, renaming을 말하며, change는 segment transforming, segment visibility control, segment highlighting을 나타낸다.

동작시에, segment는 workstation과 무관한 picture storage를 형성한다. 이를 통하여 set-up된 workstation independent storage라 부르는 special workstation을 통해서, segment는 다른 workstation으로 insert되거나 transfer될 수 있다.

5.5 Input

operator가 어떤 action을 취함으로써 입력되는 graphic 정보는 통상 logical input value라 지칭되는 여섯가지의 input data type으로서 나타내지며, 이러한 device를 logical input device라 부른다. display surface상에서 prompt나 echo와 같은, input action에 대한 결과는 각각 logical input device에 대해서 GKS에 의해 제어된다.

a) LOCATOR : WC내에 위치와 normalization transformation number를 제공
b) STROKE : WC내에 point들과 normalization transformation number를 제공
c) VALUATOR : real number를 제공
d) CHOICE : non-negative integer, 0는 "no choice"를 나타낸다.
e) PICK : pick status, segment name, pick identifier 등을 제공. segment 밖에 primitives들은 pick될 수 없다.
f) STRING : character string

각각의 logical input device는 3가지 mode로 동작되며, 이를 operating mode라 하고, SET〈input class〉MODE function에 의해 지정된다. operating mode에는 REQUEST, SAMPLE, EVENT의 3가지가 있다.

5.6 GKS attributes

display상에 나타나는 picture (output primi-

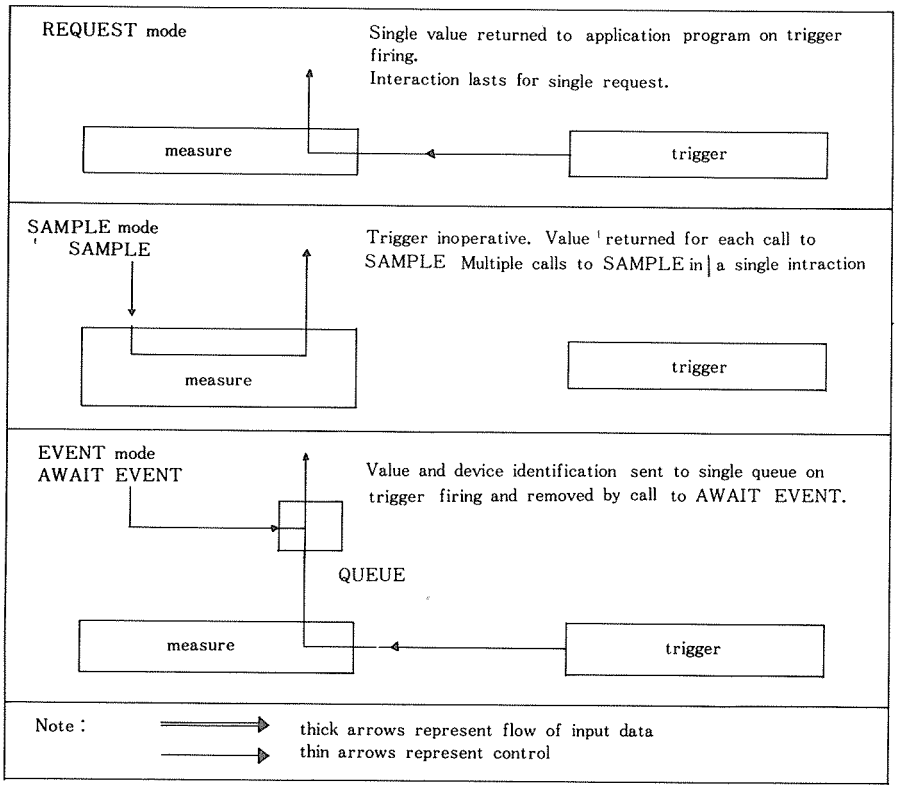


그림 6. The relationship between the measure and trigger for different operating modes, illustrated for a single logical input device

tives, segments, prompts and echo type of input device)를 제어하는 attribute는 primitive attributes와 workstation attributes로 나누어진다. primitive attributes는 text에 대한 character height나 fill area에 대한 pattern size와 같은 모든 geometrical aspects를 포함한다. primitives의 non-geometric aspects는 두가지 방법 중에 하나로 primitive attributes에 의해 제어된다. workstation dependent representation(set of value)을 가리키는 index에 의한 모든 non-geometric aspects를 specify하기 위하여 single attribute가 사용되며, workstation과 별도로 각각의 primitive에 대한 non-geometric aspect를 specify하기 위해 하나의 attribute가 사용된다. GKS에서는 전자의 방법을 Bundled specification, 후자를 Individual specification이라 한다.

workstation attributes는 non-geometric aspects의 bundled specification에 의해 사용되는 index들에 의해 point되는 workstation상에 실제 representation을 포함한다. 예를 들어, 각각의 polyline에 대한 represention(또는 bundle)은 value of linetype, linewidth scale factor, colour index 등을 포함한다. workstation attributes는 dynamic하게 reset될 수 있다.

segment를 control하는 segment attributes는 segment transformation, visibility, highlighting, segment priority and dectability 등이 있다. segment attributes는 dynamic하게 reset될 수 있으며, manipulation시에 feedback을 위한 base가 될 수 있다.

logical input device에 대한 attributes는 input device setting시, 또는 initialization시에 줄 수 있으며, initial value, prompt나 echoing technique, echo를 위한 screen area 지정 등이 있다. input device시에 operating mode와 echo에 대한 on-off switch가 선택될 수 있다. input device에 대한 operating mode란 누가(input de-

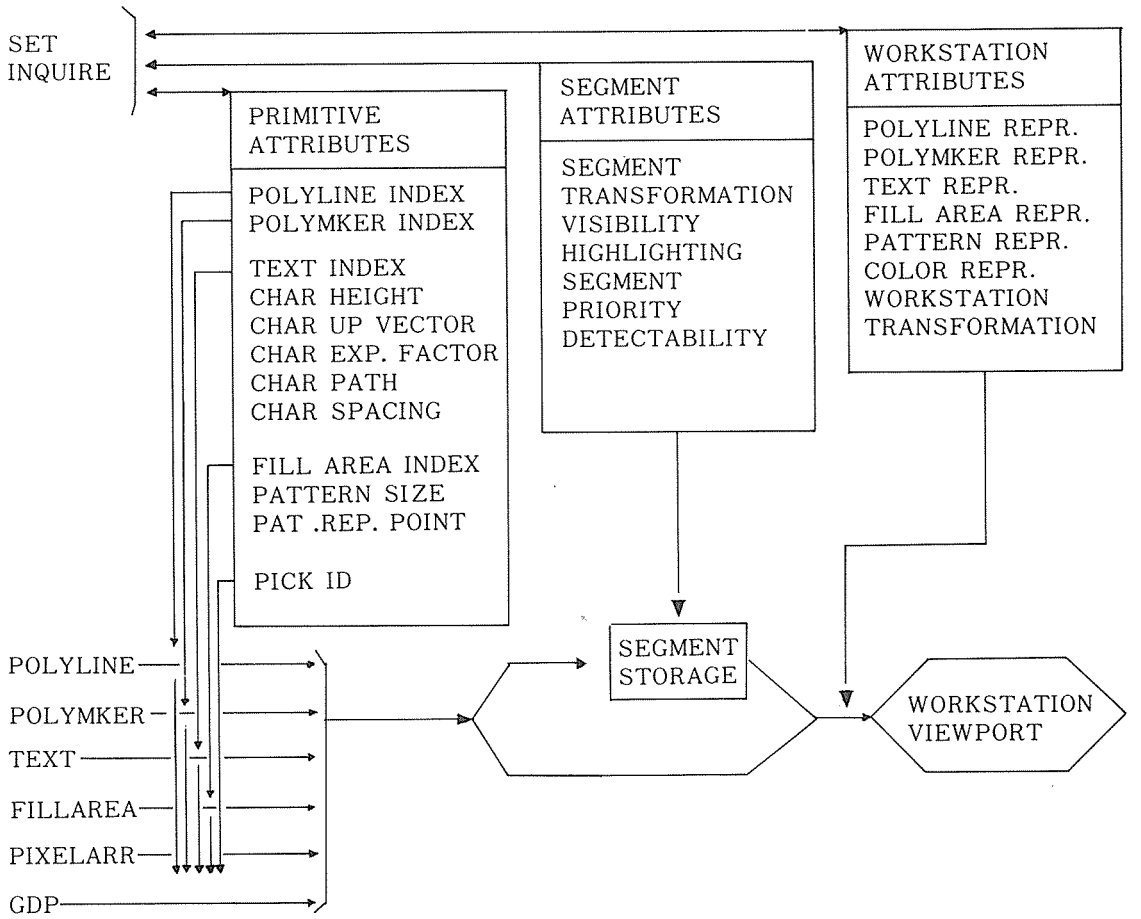


그림 7. Binding of attributes

vice 또는 application program) initiative를 가지고 있느냐를 지정하는 것으로, SAMPLE, REQUEST, EVENT 등이 있다. GKS에는 다섯개의 operating states를 가지고 있으며, 동작시에 이들 중에 하나로 수행된다. 각 state는 state내에서 허용되는 GKS function들과 state variable들로 구성된다.

5.7 GKS metafile

GKS에서는 graphical정보를 system에 저장하기 위한 interface를 제공하는데, 이를 graphics metafile이라 부르며, GKS metafile output workstation, GKS metafile input workstation으로 구성된다. 이 metafile은 다음과 같은 용도로 사용될 수 있다.

a) system사이에 graphics 정보의 이송
b) 저장매체 사이에 graphics정보의 이송
c) GKS application에서 다른 application 으로 graphics정보의 이송
d) non-graphics정보의 혼합 구성

5.8 GKS levels

GKS는 많은 응용분야에서 사용이 가능하도록 설계되어 있다. 따라서 GKS는 기능적인 면이나 input, output implementation level 등 표준사양대로 모든 것을 다 수용할 필요는 없다.

GKS의 기능적인 면은 다음과 같이 분류된다.

a) output
b) input
c) workstation의 수
d) attributes
e) segmentation

GKS의 level은 output level, input와 그외 level 두가지로 분류된다.

a) output level
m : minimal output

0 : all primitives and attributes
1 : basic segmentation and full output
2 : workstation independent segment storage

b) Input level
a : No input
b : REQUEST input
c : FULL input

7. NAPLPS 분석

캐나다의 Telidon에 기초를 둔, 통신 시스템의 picture description이 새로운 표준화로 고려중이다. 이는 Tektronix, Intel, DEC 등이 지지하고 있으며, Raster graphics device를 위해 고안되었다. NAPLPS는 화상정보의 전달 및 저장을 위하여 정보를 압축시키는 방법을 표준화하고 있다.

동작은 display device에, alphageometrics라는 graphic coding을 사용하여 Picture Description Instructions (PDIs)을 전송하여 로컬 마이크로프로세서에서 이를 해독하여 화상을 그린다. 또한, 동작모드는 alphanumeric, block-graphics 및 high resolution memory mode가 선택 사용되며 텔레비젼 케이블이나 전화선을 이용하여 two-way통신이 이루어진다.

현재, NAPLPS는 프랑스 Antiope의 alpha-mosaic data를 처리할 수 있으나, 영국 Prestel과는 호환성이 없다. 또한 NAPLPS는 로컬 프로세싱 능력이 있으므로, color mapping, macro-picture description instructions, protected fields 및 multi-window기능 등을 추가 발전시키는 것이 가능하다.

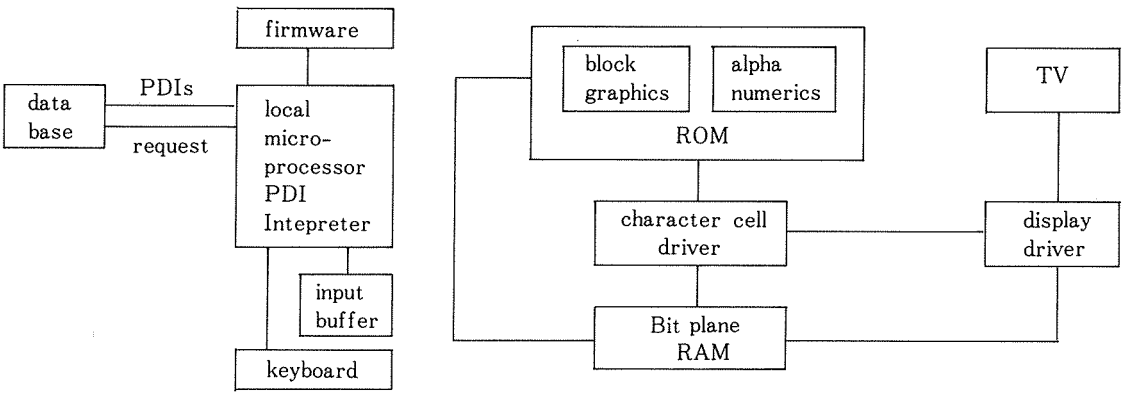


그림 8. NAPLPS system의 구성

※ 참고문헌

1. "Graphics Standards," ACM Computing Surveys, Vol. 11, NO, 4, Dec., 1978
2. Peter R. Bono, "GKS-The First Graphics Standard," IEEE Computer Graphics & Application, July, 1982, pp. 9-23
3. Status Report of the Graphics Standards Planning Committee, Computer Graphics, ACM SIGGRAPH, Vol. 13, No. 3, Aug., 1979
4. Ware Myers, "Computer Graphics: The Human Interface," IEEE, Computer, June, 1980, pp. 45-54
5. Fred E. Langhorst, "Realizing Graphics Standards for Microcomputers," Byte, Feb., 1983, pp. 256-268
6. David H. Strayer, "Hoisting the Color Standard," Computer Design, July, 1982, pp. 123-130
7. Fred Langhorst, "Working Toward Standards in Graphics," Computer Design, July, 1982, pp. 177-182
8. Chris Baily, "Graphics Standards are emerging slowly but surely," Electronic Design, Jan. 1983, 103-110
9. "The GKS Impact Graphics Standardization," Computer Graphics World, Nov., 1982, pp. 45-51