

ISO グラフィクス・メタファイル標準VDM

川合 慧 (東京大学理学部)

1. はじめに

本稿では、ISO における一連のグラフィックス規格案のうち、図形・画像の蓄積や通信に主眼点を置いた Virtual Device Metafile (VDM) について概説する。VDM の規格案作りは ANSI が担当しており (X3H33)、現在最終段階に入っている。本稿は、dpANSI の案となっており最新の内容をもとにして、その概要を抜粋要約したものである。また、これまでの VDM 規格案との相違点にも触れる。なお、グラフィックス標準化活動全体の現在に至るまでの経緯や、その中における VDM の位置づけなどについては、他の解説を参照された。

2. VDM 概説

2.1 目的

規格案の中では、VDM を定義する目的として、以下のような項目をあげている。

- * グラフィクス・パッケージ作成者に対する、データ・インタフェースの標準の制定
- * グラフィクス関連の装置およびシステム間の、図形データ転送の標準の制定
- * 端末固有の機能にアクセスするための標準の制定

これからわかるとおり、VDM の規格は、図形データの定義法と、その正確な意味づけを与えることを意図している。したがって、VDM の作成 (generation) や解釈 (interpretation) の具体的な方法は定義されない。この点では、グラフィックス標準である GKS の規格と同じ性格を持っている。しかしながら、GKS が対話型式も含めた極めて広い範囲のグラフィックスを指向しているのに対して、VDM はその守備範囲を、対話を含まない図形記述に限っている。この設計方針から、以下のような VDM の特徴づけがなされた。

- (1) 使用する概念や技術用語については、可能な限り GKS と一致させる。
- (2) GKS で間合せ (engulry) によってのみ実現できる事項については、GKS にはない機能を導入してこれを実現する。
- (3) 蓄積や通信のための、符号化 (encoding) の標準的な方法を定めておく。

VDM に関連した概念として、Virtual Device Interface (VDI) がある。VDI は、グラフィックス・システム全体のうちの、装置独立な部分と装置依存の部分の間のインタフェースと定義されている。これも、同じ ANSI の X3H33 によって規格作りが進められていたが、GKS と VDM といった“上位の”規格の作業の終了後に検討を再開することになっている。ちなみに、VDM の規格案の中では、諸 VDM の定義がつかのようになされている。

『グラフィックス・データと制御情報の保持および転送のための VDI レベルにおけるメカニズム』

グラフィックス・システムにおける VDM の役割を図に示す。

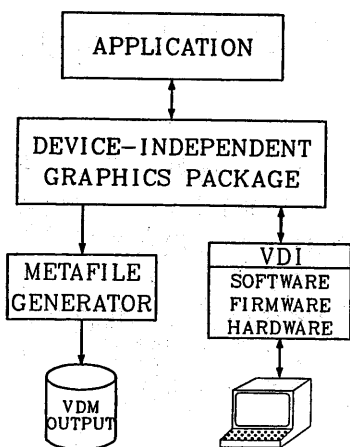


図1. 一般のシステムと VDM

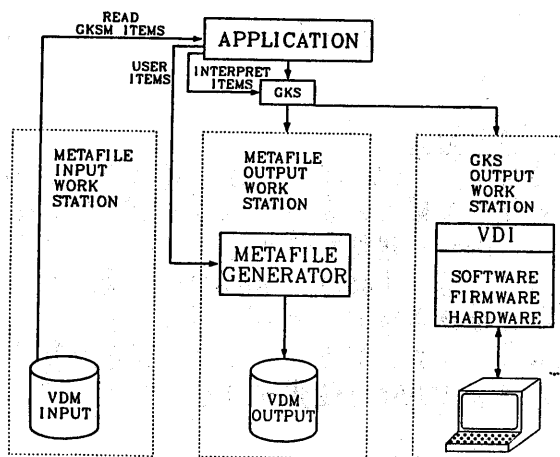


図2. GKS と VDM

図1は、一般のグラフィックス・システムでの VDM の作成を示す。作成するレベルが、右側の VDI と同じになっていることがわかる。図2は GKS と VDM の関係を示す。VDM は、GKS のメタファイル用ワークステーションに関連づけられるようになっている。すなわち、VDM を作成するには、作成用のサブシステムを組み込んだワークステーションを用意しておけばよい。読取りの方は、メタファイル入力ワークステーションから、GKS の命令である READ GKSM ITEMS, INTERPRET ITEMS によって行なう。GKS 以外のシステムでは、VDM 解釈プログラムが VDM を読取り、表示出力を行なうことになる (図3)。

2.2 基本概念

(1)座標系 VDM では、Virtual Device Coordinate (VDC) とよばれる、ただひとつの座標系を用いる。座標変換の機能はない。VDC には整数型と実数型がある。各 VDM ごとに、このどちらかの型が選ばれる。

VDC の中に、図形を定義する範囲として VDC Extent を設定する。これは、最終的な表示画面における、左下と右上の2点の座標で指定する。この際、双方の点の座標値の大小関係は任意であり、VDC の軸方向と表示面上での軸方向の対応づけが、VDC Extent の設定によっても可能である。たとえば、左下 = (0, 4095), 右上 = (4095, 0) とすれば、左上も原点とする表示面に適合させられる (図4)。また、この左下・右上の指定によって、角度の正負や文字列の方向などの定義がなされる。

VDC の中には、クリッピングを行なうための Clip Rectangle が指定できる。

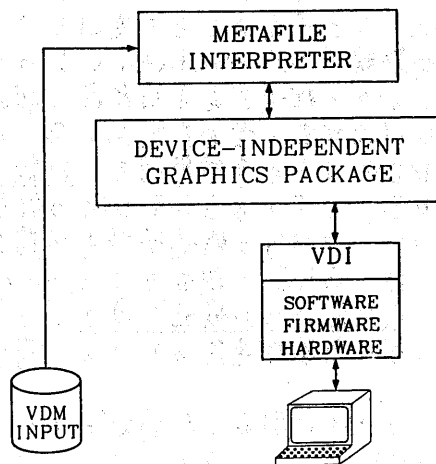


図3. VDM の解釈と出力

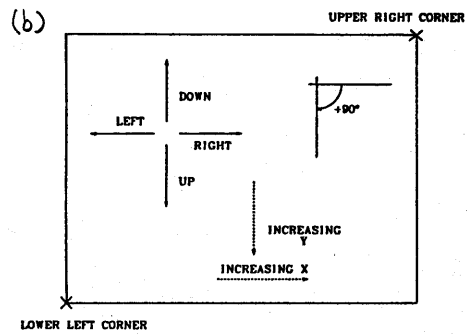
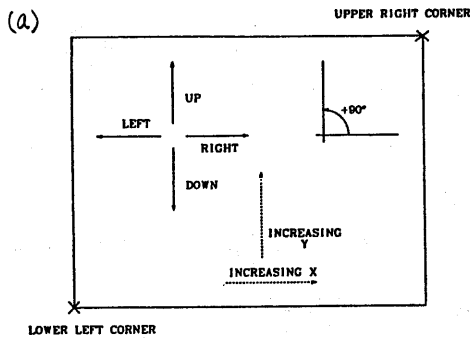


図4 VDC Extent の指定 (a)... 左下原点, (b)... 左上原点

(2) 図形要素 VDM における図形の要素は以下のとおりである。

POLYLINE ... 折れ線

POLYMARKER ... マーカの集合

POLYGON ... 閉多角形領域

CIRCLE ... 円領域

ARC ... 円弧

ARC CLOSE ... 閉じた円弧領域

TEXT ... 文字列

APPEND TEXT ... 文字列の続き

CELL ARRAY ... ぬりつぶし領域

これらのうち、CIRCLE, ARC, ARC CLOSE, APPEND TEXT の各要素は、GKS にはない。また POLYGON は、GKS の FILL AREA に対応する。ARC 等は、端末ごとのサポートの一般性から導入された。APPEND TEXT については後述する。

(3) 属性 図形要素の属性は、GKS のそれとほぼ同じである。要素ごとに複数の属性を一括して指定する機構 bundle と、それを使うか使わないかを切り換えるフラグ Aspect Source Flag のメカニズムも、GKS のそれを踏襲している。以下に、GKS との主な相違点を示す。

(i) 入力がないので、属性 PICK-ID がない。

(ii) 線幅やマーカの大きさについて、絶対的な値による指定も可能とする。

(iii) 色についても、色インデックス以外に、絶対値による指定も可能とする。

(iv) 領域要素 (POLYGON, CIRCLE, ARC CLOSE) については、境界線 (perimeter) のための属性を追加する。

1. Attributes (bundlable and individual)

POLYLINE bundle for **POLYLINE, ARC**
 LINE TYPE solid, dash, dot, dash-dot, dash-dot-dot, etc.
 ---- WIDTH abs. width or scale factor
 ---- COLOR color index/value

POLYMARKER bundle for **POLYMARKER**
 MARKER TYPE dot, plus(+), asterisk, circle, cross(X)
 ---- SIZE abs. size or scale factor
 ---- COLOR color index/value

FILL AREA bundle for **POLYGON, CIRCLE, and ARC CLOSE**
 INTERIOR STYLE hollow, solid, hatch, pattern
 FILL COLOR color index/value
 HATCH INDEX pointer into the hatch table of the device
 PATTERN INDEX pointer into the pattern table
 PERIMETER TYPE same
 ---- WIDTH as
 ---- COLOR LINE

TEXT bundle for **TEXT, APPEND TEXT**
 TEXT FONT INDEX font index
 ---- PRECISION string, character, stroke
 ---- COLOR color index/value
 CHARACTER SPACING fraction to CHAR-HEIGHT
 ---- EXPANSION FACTOR

Individual Attributes

PATTERN SIZE deltax and deltax
 ---- REFERENCE POINT VDC point
 CHARACTER HEIGHT VDC length
 ---- ORIENTATION UP- and BASE_vector
 ---- PATH right, left, up, down (H, V, CH, CV) where
 TEXT ALIGNMENT
 H=left, center, right,
 normal horizontal, continuous horizontal
 V=top, cap, half, base, bottom,
 normal vertical, continuous vertical
 CH, CV real values
 when "normal" the alignment is
 path-dependent as follows
 path == right left up down
 horizontal left right center center
 vertical base base base top

(v) 文字列の属性として, Character Up Vector (GKS) と Character Orientation という2個のベクトルに拡張し, Text Alignment にも新しいものをつけ加えた。

(4) 文字列

文字列は, 要素となる文字の属性と, 文字の配列方法に関する属性との両方を持つ。したがってその指定法も複雑なものとなり, GKS, VDM 双方において, 最もこみ入ったモデルも持つ出力基本要素である。図5に, GKSにおける文字列属性の指定方法を示す。

図5(c)で示されているように, 文字列の出力は, 基準位置(x, y)と文字の列による行なわれる。したがって, 複数個の文字列を一直線上に並べたり, 行単位に配置したりする場合には, ひとつ前の文字列とうまくつながるような, 新しい基準位置を求めなければならない。GKSでは, これを手える enquiry, "INQUIRE TEXT EXTENT" を用意しており, 文字列を囲む長方形と, 次に指定すべき基準位置とがわかる仕組みになっている。ところが VDM では問合せを行なわないこととしたので, 文字列の整理のために, GKS にはない機構を導入した。そのひとつが

図6に示す APPEND TEXT で, "直前のTEXTの続き" という意味を持たせ, かつ, 途中で属性の変更ができるようにした。この機能は, 1行の中の一部だけ色や大きさや文字種を変えるのに有効である。

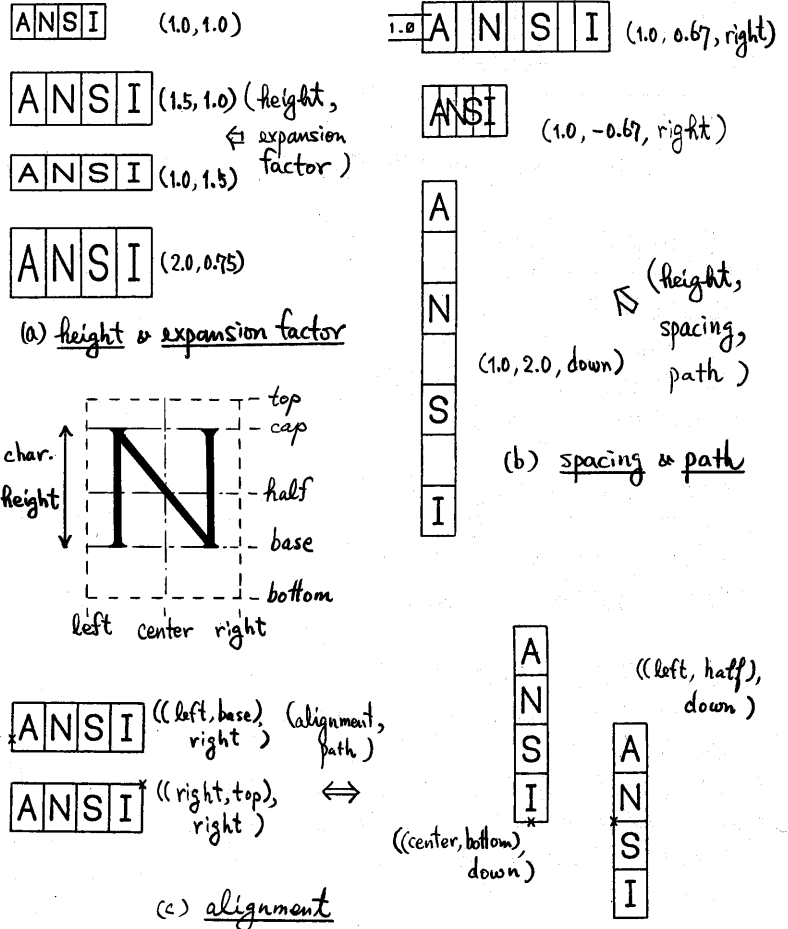
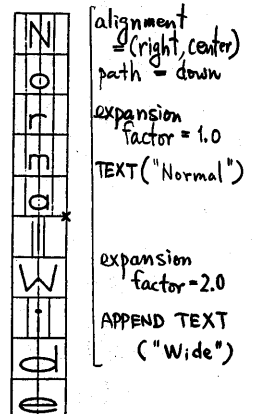


図5. 文字列の属性 (GKS)



alignment = (center, cap)
path = right
height = 2.0
TEXT ("Big")
height = 1.0
APPEND TEXT ("Little")

図6. APPEND TEXT



文字列の行単位の整列のためには、配置のための属性 text alignment に新しい値

continuous horizontal/vertical

と、それぞれの程度を表わすふたつの実数値がつけ加わった。この値は、文字列を囲む長方形のどの場所を基準位置に合わせようかを示すもので、長方形の縦および横の長さを単位として指定する(図7(a),(b))。したがって

continuous horizontal については

0.0 が left, 1.0 が right と、

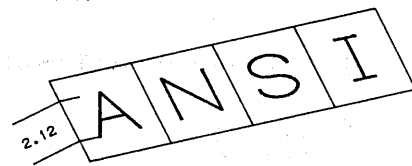
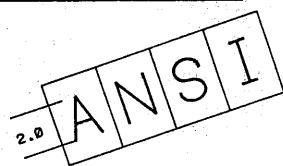
continuous vertical については

0.0 が bottom, 1.0 が top と

それぞれ等価である。この属性を使うことにより、同一の基準位置も指定した、行単位の整列が可能となる(図7(c),(d))。

GKS では、もとの座標系では立体の文字しか許さなかったが、縦横の倍率が異なる正規化変換が施さされると、斜体文字も扱われなければならない。VDM には正規化変換がないので、この“斜体”文字を扱うために、文字の上方向(char. up)と直角でないことも可能なベクトル char. base vector を導入した。このふたつのベクトルは、まとめて char. orientation として指定する(図8)。

以上で示した APPEND TEXT と新属性値 continuous は、GKS にはないものであるが、文字列の大域的な扱い方法のひとつとして注目される。



(a) alignment = (continuous horizontal, base, 0.25, 0)

(b) alignment = (continuous horizontal, continuous vertical, -0.25, -0.25)

(c) alignment = (left, continuous vertical, 0, 1)
"ANSI"

(d) alignment = (left, continuous vertical, 0, 2.5)
"X3H3"

(e) alignment = (continuous horizontal, top, 0, 0)
"ANSI"
(continuous horizontal, top, -2.0, 0)
"X3H3"

図7. Continuous Alignment

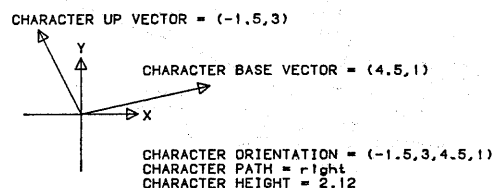
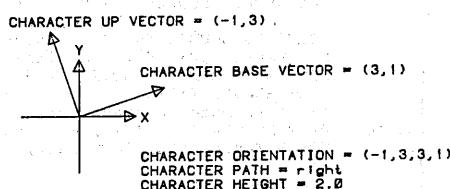


図8. Character Orientation

(5) 値指定のモード VDM では、出力図形の絶対的な大きさを指定することも可能とするために、値の指定にモードを設けている。ここからは、VDM の中に複数個含まれ得る図形のひとつひとつについて、別々に指定する。

(i) SCALING モード: VDC が実際の長さ(mm)に対応するか、抽象的なものかの指定。前者の場合は、1 VDC 単位が何 mm に当たるかの指定。

(ii) COLOR 指定モード: 色を RGB の値で表わすか、指標値で表わすかの指定。

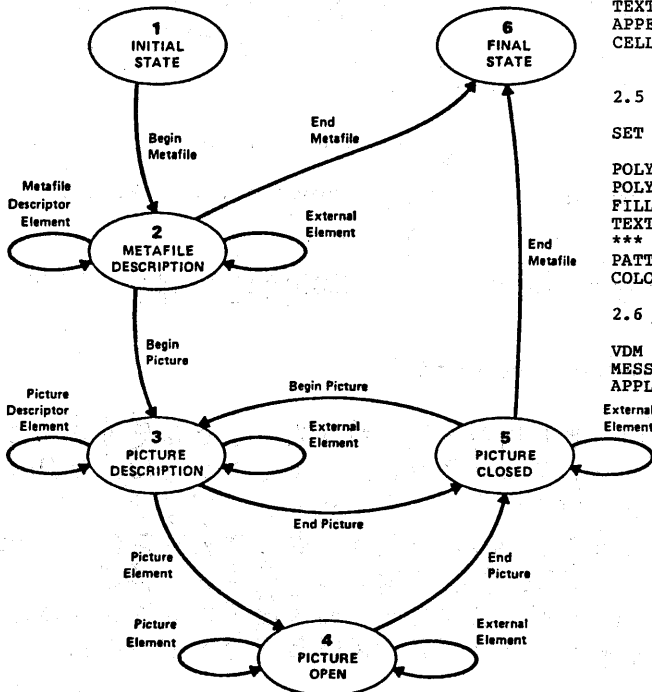
(iii) WIDTH/SIZE 指定モード: 中や大きさの指定が直接値か倍率かの区別。

2.3 要素と構成

VDMの要素を右に示す。Descriptor要素は、あるVDMに関して、その記述法や名前、精度、使用文字セットなどを宣言するためのものである。とくにVDM ELEMENT LISTは、以降に続くVDMに現れ得る要素も列挙するもので、解釈ループに“心構え”をさせるための要素である。Control要素は、VDM中の各部分の区切りを示すが、主な役目である。図9に、Control要素ほかによる、VDMの状態遷移を示す。VDM全体は、1個のディスプレイと0個以上の図とからなり、1個の図は、1個の図ディスプレイと図形定義要素列とからなる。

Picture Descriptor要素は、この図形に関する値指定のモードを設定する。Graphical要素とAttribute要素が、実際の図形記述のための要素である。ほかの要素のうち MESSAGE 要素の action パラメータは、オペレータが指定するまで解釈を休止するためのものである。

図9. VDMの状態遷移図



2. Metafile Elements

2.1 Descriptor Elements

VDM VERSION	int
VDM DESCRIPTION	string
VDC TYPE	int/real
NON-VDC INTEGER PRECISION	
NON-VDC REAL PRECISION	
INDEX PRECISION	
COLOR PRECISION	
COLOR INDEX PRECISION	
MAXIMUM COLOR INDEX	index
VDM ELEMENT LIST	list of elements in the VDM
VDM DEFAULT REPLACEMENT	
FONT LIST	strings
CHARACTER SET LIST	

2.2 Control Elements

BEGIN METAFILE	string
END METAFILE	
BEGIN PICTURE	on/off, string, view-surface-color
BEGIN PICTURE BODY	
END PICTURE	
LOCAL BACKGROUND COLOR	on/off, background-color
INTEGER VDC PRECISION	
REAL VDC PRECISION	
VDC EXTENT	
CLIP RECTANGLE	xmin,xmax,ymin,ymax
CLIP INDICATOR	on/off

2.3 Picture Descriptor Elements

SCALING MODE	abstract/metric
COLOR SELECTION MODE	indexed/direct
LINE WIDTH SPECIFICATION MODE	absolute
MARKER SIZE SPECIFICATION MODE	or
PERIMETER WIDTH SPECIFICATION MODE	scaled

2.4 Graphical Elements

POLYLINE	point list
POLYMARKER	point list
POLYGON	point list
CIRCLE	center, radius
ARC	3 points
ARC CLOSE	3 points, pie/chord
TEXT	point, final/not_, string
APPEND TEXT	final/not_final, string
CELL ARRAY	xmin,xmax,ymin,ymax, M,N, color list

2.5 Attribute Elements

SET ASPECT SOURCE FLAGS	list of (ASftype, individual/bundled)
POLYLINE BUNDLE INDEX	index
POLYMARKER BUNDLE INDEX	index
FILL AREA BUNDLE INDEX	index
TEXT BUNDLE INDEX	index
*** Attribute settings for all aspects ***	
PATTERN TABLE	index, M,N, color list
COLOR TABLE	starting index, color value list

2.6 Other Elements

VDM ESCAPE	index, data list
MESSAGE	action (on/off), string
APPLICATION DATA	identifier, string

↑ 表中のパラメータ
 int: 整数 real: 実数
 string: 文字列 index: 指標
 on/off: フラグ point: 点
 xmax 等: VDC値
 M,N: 整数

3. VDMの符号化

3.1 符号化標準

VDMの規格案には、標準的な符号化として、文字およびバイナリの符号化が含まれている。これは、GKSにおける language binding に相当する。

3.2 文字符号化

この符号化は、ISO 2022-1982のコード拡張法に適合するように決められている。制御用の文字コードは、図10に示す制御用のCOセットのうち、コード拡張用の Shift-in(0/14), Shift-out(0/15), Escape(1/11) および、VDM要素やパウ×タの区切りのための Inf. separator one (1/14), two (1/15) の5個である。

VDM要素は、

図10に示した
96文字セット

によって符号化される。ある要素は、要素コード1ないし2文字と、パウ×タとかなる。要素コードの区分けは

3欄 図形

4,5欄 属性

7欄 ESCAPE

2欄 その他

となっている。パウ×タはすべて可変長で、たとえば整数は

```
* 1 d d d d d d
* 1 d d d d d d
...
* 0 i s d d d d
```

という形式である。

(*: 任意, d: 0/1, s: 符号)

また、データ量の特に多い CELL ARRAY のパウ×タについては、run length 形式と普通の形式を用意し、混在を許している。

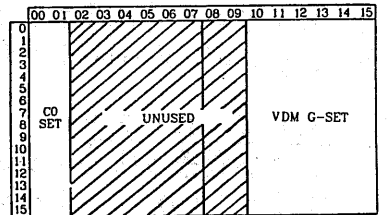
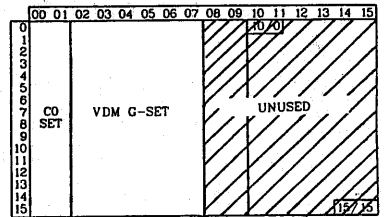
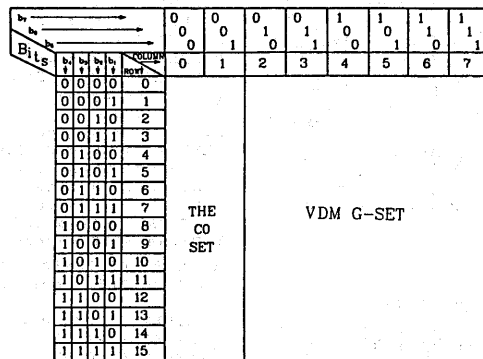


図10. VDM 文字セット 7ビット(左), 8ビット(右)

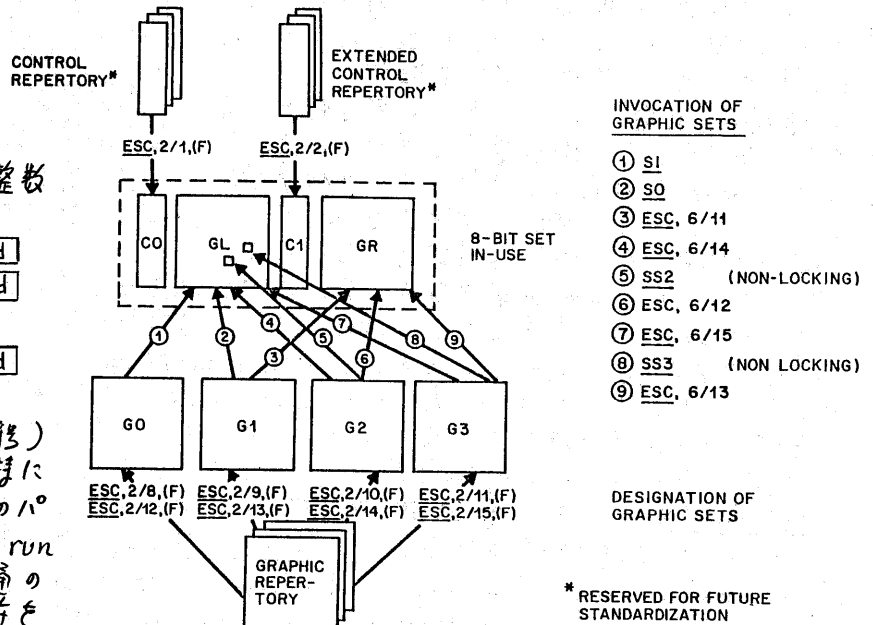


図11. 文字コード拡張法

要素コードの一覧を右に示す。先頭のふたつについては、もう1文字(3/.)を使って、個々の要素を指定する。

3.2 バイナリ符号化

この符号化は、VDMの各要素を表現するのに必要な、最小限の情報の形式を示したものである。仮想的なVDM解釈マシンの機械語の形式と思えばよい。

機械語の命令部分は、要素の大分類4ビット、小分類7ビット、およびパウ×ラ長の指定5または20ビットからなる1または2語の形式である(1語=16ビット)。パウ×ラは、整数(符号付または無し)8, 16, 24, 32ビット、文字8ビット、実数32, 64ビットのいずれかで、詰め合わせて収容される。実数は、IEEE floating point standardの表現法を使用する。

4. 規格案の移り変り

VDM規格案の内容の進展は以下のとおり。

(i) 1982年5月版 ... GKSとの整合性はまだ考慮されていない。したがって、bundleの概念がない。indow-newpor変換があった。VDCの逆向き指定(図4(b))は不可。図形要素としてFILL AREAがなく、かわりに長方形があった。拡張性のために次元数(2/3)を指定した。

(ii) 1983年3月版 ... GKSとの整合性をとる努力がなされ、bundleが導入されたが、ASFは各Bundleにひとつだけ設けた。座標変換は無くなった。FILL AREAが導入されたが、PATTERN REFERENCE POINTは指定できなかった。TEXT ALIGNMENTのnormal値がなかった。各PICTUREの先頭で表示画面はクリアされた。符号化の章が充実したが、現在のものとは大中に異なっていた。

(iii) 1983年8月版 ... 属性ごとのASF, PATTERN REFERENCE POINT, ALIGNMENTのnormal値などがGKSと同じになった。CHARACTER ORIENTATIONが導入された。閉領域の境界線の表示属性を独立させた。各PICTUREごとの画面クリアは、予らなりことになった。Picture Descriptor要素というクラスができた。次元数指定は予めた。

(iv) 1983年11月版 ... 新要素BEGIN PICTURE BODYが追加され、ESCAPE, MESSAGE, APPLICATION DATAの形式が定まった。符号化の方法が大中に変更された。

* METAFILE DESCRIPTOR opcode	2/0
* PICTURE DESCRIPTOR opcode	2/1
BEGIN-END METAFILE opcode	2/2
BEGIN PICTURE opcode	2/3
BEGIN PICTURE BODY opcode	2/4
END PICTURE opcode	2/5
INTEGER VDC PRECISION opcode	2/6
REAL VDC PRECISION opcode	2/7
VDC EXTENT opcode	2/8
CLIP RECTANGLE opcode	2/9
CLIP INDICATOR opcode	2/10
LOCAL BACKGROUND COLOR opcode	2/11
MESSAGE opcode	2/12
APPLICATION DATA opcode	2/13
POLYMARKER opcode	3/0
POLYLINE opcode	3/1
POLYGON opcode	3/3
CELL ARRAY opcode	3/3
TEXT or APPEND TEXT opcode	3/4
CIRCLE opcode	3/5
ARC opcode	3/6
ARC CLOSE opcode	3/7
POLYMARKER BUNDLE INDEX opcode	4/0
MARKER TYPE opcode	4/1
MARKER SIZE opcode	4/2
MARKER COLOR opcode	4/3
POLYLINE BUNDLE INDEX opcode	4/4
LINE TYPE opcode	4/5
LINE WIDTH	4/6
LINE COLOR opcode	4/7
FILL AREA BUNDLE INDEX opcode	4/8
INTERIOR STYLE opcode	4/9
FILL COLOR opcode	4/10
HATCH INDEX opcode	4/11
PATTERN INDEX opcode	4/12
PATTERN SIZE opcode	4/13
PATTERN REFERENCE POINT opcode	4/14
PATTERN TABLE opcode	4/15
PERIMETER TYPE opcode	5/0
PERIMETER WIDTH opcode	5/1
PERIMETER COLOR opcode	5/2
TEXT BUNDLE INDEX opcode	5/3
TEXT FONT INDEX opcode	5/4
TEXT PRECISION opcode	5/5
CHARACTER EXPANSION FACTOR opcode	5/6
CHARACTER SPACING opcode	5/7
TEXT COLOR opcode	5/8
CHARACTER HEIGHT opcode	5/9
CHARACTER ORIENTATION opcode	5/10
CHARACTER PATH opcode	5/11
TEXT ALIGNMENT opcode	5/12
CHARACTER SET INDEX opcode	5/13
COLOR TABLE opcode	5/14
SET ASPECT SOURCE FLAGS opcode	5/15
(Reserved for 2-character std. opcodes)	7/12
VDM ESCAPE opcode	7/13
(Reserved; not to be used)	7/14

参考文献

- dpANS Information Processing Computer Graphics VDM Functional Description (Nov. 1983)
- Draft Proposed American National Standard for the VDM, ANSI X3H33
82-15 R1 (May, 1982), 83-16 (March 1983), 83-15 R1 (August 1983)
- Graphical Kernel System (GKS) Functional Description V.7.2, ISO/TC97/SC5/WG2 (Dec. 1982)