ＧＫＳと階層化メニュー

3Q-9

小野眞、清水和哉
日本アイ・ビー・エム株式会社、サイエンス・インスティチュート

はじめに

人間にとって使いやすい計算機環境を提供するために，様々な「人間－機械インターフェイス」が提案されまた実現されている．一方近年のワークステーションの増加にともない，個々の「人間－機械インターフェイス」は入出力装置に対しそれぞれ固有のプログラム・インターフェイスを持って実現されているのが現状である．

図形画像表示を行なうコンピュータグラフィックスの分野では，人を含めた資源の利用と可搬性に支障が生じたため早くからこのプログラム・インターフェイスの標準化が考えられており，そのひとつにＩＳＯ規格による Graphical Kernel System(ＧＫＳ）と呼ばれるものがある．

このＧＫＳは表示のみならず入出力全般におよぶ規格となっているが，元来ホストーワークステーションといった構成を基に考えられたため，現在要求されている環境をＧＫＳを用いて実現するには困難がある．本稿ではその問題点を検討するとともに新たな仕様を提案する．

モードのない環境

「利用者が今どのモードにいるのか，またそのモードでは何を行なうことができるのか，他のモードとはどのようにして移り変われるのか」．このような「モード」という概念は，利用者に対し混乱を与え，システムを使いにくくしている原因であると指摘されている(1)．

しかし本当にモードをもたないシステムを設計できるのだろうか．利用者が同時に制御可能な（または把握することができる）操作の数には制限がある．一方利用者の要求はかなり複雑かつ大きなものであることが多い．そのギャップを埋めるためにしばしば用いられる方法のひとつに，「シフトキー方式」がある．人が制御できる鍵盤の数には制限があるが，シフトキーを押している間は通常とは異なった意味をひとつのキーに与えることで，多くの入力を行なうことができる．このシフトという状態もある意味では「モード」に等しい．先に指摘された「モード」との大きな違いは，「利用者がシフト状態にいるのは，［利用者が意識して］シフトキーを押し続けている間のみである」点であろう．

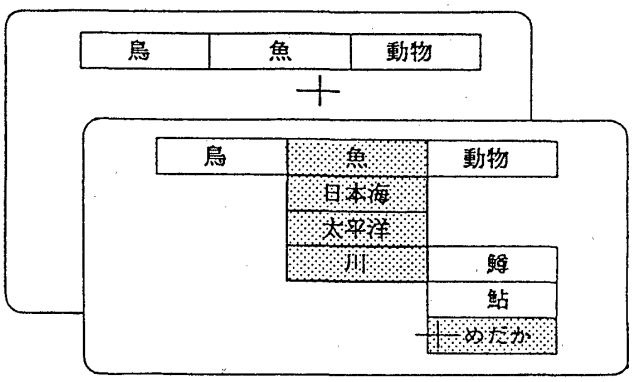

図１　階層化されたメニュー

たとえば，図１は階層をもつメニューの一例である．利用者の操作（最初のメニューの中より次のメニューを導きだす項目を選択すること）によってシフトインの状態（２番目のメニューを表示して，その中から選択を行なう）を作り出していることがわかる．

ＧＫＳを用いた場合

まず最初にＧＫＳを用いて前述の「階層化メニュー」を実現する方法を考察しよう．このようなメニューの表示をＧＫＳのイベントモードと入力促進／エコーを用いて実現することは困難である．通常はサンプル入力を用いて実現されると思われる．メニュー用の選択値入力装置(choice device)S_1，S_2を定義する．さらにこのメニュー入力を開始する制御用として，選択値入力装置S_0を定義する．S_0をサンプルモードで起動し，入力された値が第一のメニューを開始する値になるまで読み続ける．つぎに入力装置S_1もサンプルモードで入力を開始する．その後S_0の入力値が変化しないまま，S_1の中で第二のメニューを開始する項目が選択された場合，入力装置S_2もまたサンプルモードで入力を開始する．S_2の中で選択が終了した場合，S_2・S_1のモードをリクエストモードに戻す．しかしながら以上の方法には次の三つの問題点がある．

GKS and Hierarchical Menu
Makoto ONO, Kazuya SHIMIZU
Science Institute, IBM Japan Ltd.

(1) これらの処理の多くはワークステーション内でできることである．
(2) 応用プログラムの制御がかなり複雑になる．
(3) あきらかに効率がわるい．

これらは，ＧＫＳのメジャープロセスの制御が応用プログラムにしかできない点に起因するものと考えられる．我々は，このメジャープロセスを利用者の操作からも制御できる，新たな仕様を提案する．

メジャープロセス制御

「ワークステーション状態リスト」の中に各論理入力装置ごとに「メジャープロセス制御状態欄」を導入する．この欄には

(i)この装置を起動する論理入力装置名
(ii)入力値の範囲(value range)

が記述されている．入力値の範囲とは，たとえば位置入力装置(locator device)であればその座標が[Xmin, Xmax] [Ymin, Ymax]で定義される矩形域を指し，また選択値入力装置であれば選択値集合の部分集合を指す．(i)の論理入力装置が(ii)で指定された範囲に入ったときこの論理入力装置は起動されるものとする．

さらにこれらの論理入力装置がこの欄の内容によって制御されることを示す第４のモード，結合モード(Connected Mode)を導入する．各プロセスの制御の移り変わりをワークステーション状態リストに登録した後，必要なすべての装置を結合モードに設定する．各装置の結合関係が木状になっているものとしたならば，その根にあたる装置（これを起動するプロセスは定義されていない．先程の例では装置S_0）を任意のモードで実行させる．実行した結果，最終的に得られた値はＧＫＳ待ち行列にはいる．応用プログラムは，待ち行列から値を取り出すことによって，入力された論理装置とその値を得ることができる．

先程の階層化されたメニューをこれらの機能を使って実現すると，つぎのようになる．

(1) 選択値装置S_0およびS_1，S_2をリクエスト・モードに設定する
(2) 装置S_0およびS_1，S_2を初期化する
(3) 以下のようにメジャープロセス制御欄に登録する
　装置S_1：装置S_0の入力値が［１］のとき
　装置S_2：装置S_1の入力値が［２，３］のとき
(4) 装置S_1およびS_2を結合モードに設定する
(5) 装置S_0を任意のモードに設定する
(6) 装置S_0を実行する
(7) 入力された値をＧＫＳ待ち行列から読み取る

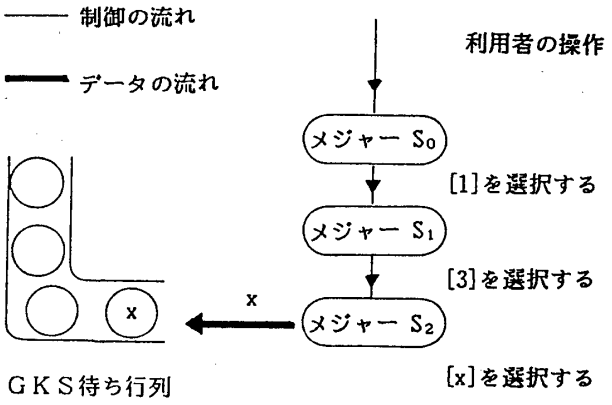

図２　利用者の操作とメジャープロセスの流れ

このときの利用者の操作とメジャープロセスの対応を図２に示す．

ＧＫＳと応用プログラムとの値の受け渡しには，最終的にどの論理入力装置から値が得られるかわからないためＧＫＳのイベント・モードの機構を利用する．一方装置S_0は任意のモードで使用できるものとした．ただしリクエスト・モードで使用したときは装置S_1またはS_2から入力された場合，ステータスNone，OKのほかにOthersを導入しこの値を返すこととする．サンプル・モードでは装置S_0の現在の値を返すものとする．その場合メジャーが中断されていたならば中断されたときの値（別のメジャーを起動する入力値範囲内の値）を返すものとする．

これらの手順によって利用者から自由にメジャープロセスを制御しつつシステムに入力することができる．

おわりに

本稿で提案した仕様は，利用者の操作からＧＫＳのメジャープロセスを制御することを可能にしたものである．我々はこの仕様に基づいてIBM 5080に対するＧＫＳの拡張インターフェースを試作中である．現在の仕様においては通常，論理入力装置の数は大変多くなるものと考えられる．しかし論理的に増えるだけであり，かならずしも同数の物理装置が必要なわけではない．今後，より高度な「人間－機械インターフェース」をも十分実現できるプログラム・インターフェースの研究をすすめるつもりである．

参考文献

(1) Larry Tesler, "The Smalltalk Enviroment", Byte, Vol.6, No.8, pp.90-147, Aug., 1981

(2) "Graphical Kernel Syatem(GKS) Functional Description", ISO 7942, Aug., 1985