

サテライト構成における

インタラクティブ・グラフィックソフトウェア

黒崎 悦明, 佐賀 六典

(沖電気工業株式会社)

1. まえがき

小型計算機を、グラフィック・ディスプレイの端末処理用として大型ホスト・コンピュータに接続したサテライト構成(あるいは、マルチ・プロセッサ構成)は、高パフォーマンスを提供し得る実用システムの構成法として最近我が国でもいくつかの例が発表されている。

特に、サテライト・システムの側にユーザのプログラミングが可能なシステム(OPEN型構成)の場合、プログラムは一般に複雑となる。さらに、アプリケーション・システムを設計していく際、作業環境の悪化、オペレーション条件の一部変更等は、頻繁かつ長期にわたってまじ、完全に安定するまでかなりの労力が費やされる。

このような不安定要因によって最も大きな影響をうけるのは、人間と一番近くに面しているサテライト・システム側におかれるプログラムである。

したがって、サテライト・システムのグラフィック・プログラムの便宜を討ることがこの様なOPEN型構成の場合、特に重要である。

本報告では、従来、我々が開発しているグラフィック・ディスプレイ・システムAGENTの概要と上述の狙いから検討をすすめている、対話型システムを設計するためのグラフィック言語GPOLについてその特徴を紹介し、システムの問題点を明白にする。

2. サテライト構成の特徴

サテライト構成は、ホストおよびサテライト側におかれるソフトウェア機能の相違により次の二つのタイプに分けられる。[1]

(1). CLOSED型

(2). OPEN型

(1). CLOSED型

サテライト・システムにおかれるPRE-DEFINEDな機能をホスト側からの指定により実行していくもので、ホストのSLAVEとなって運用される。デバイスのCOMPATIBILITYはあるがFLEXIBILITYに欠ける。

CLOSED型構成をFIG.1に示す。

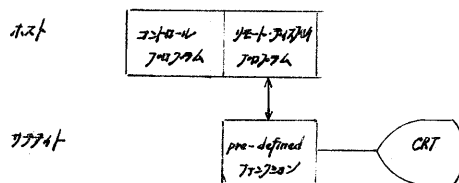


FIG.1 CLOSED型構成

(2). OPEN型

ホスト、サテライト両方にユーザのプログラムがおかれる。これら両コンピュータ間のコミュニケーションとそのコントロールはユーザの責任下におかれる。FLEXIBILITYに富み、またユーザ・アプリケーションに応じてLOAD SHAREが可能であるが一般にプログラミングは複雑となる。

OPEN型構成をFIG.2に示す。

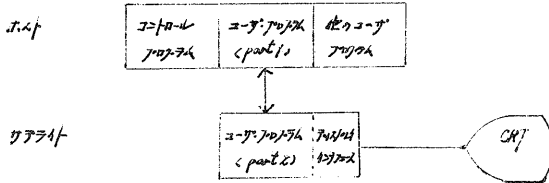


FIG.2 OPEN型構成

各々のシステム・タイプの選択は、下記の条件を充分検討した上で決められるべきである。

- (a). アプリケーションの規模
- (b). 計算機使用のコスト
- (c). 大型計算機のAVAILABILITY
- (d). リモート / ローカル
- (e). システムのコスト・パフォーマンス
- (f). 大型計算機の負荷

AGENTの場合、当面のアプリケーションは集積回路のCADである。サテライト・システムは、2400 BPSの通信回線を介してUNIVAC 1106 計算機と接続され、ホスト側リソースはデマンド・モードで使用される。さらに、サテライト側小型計算機は、グラフィック・ジョブの他にリモート・バッチ・ターミナルとしても共用される事が要求される。AGENTでは、以上の環境条件を検討した結果、OPEN型構成が採用された。

3. AGENT システムの概略

3.1. 構成

システム・ハードウェア構成をFIG.3に示す。

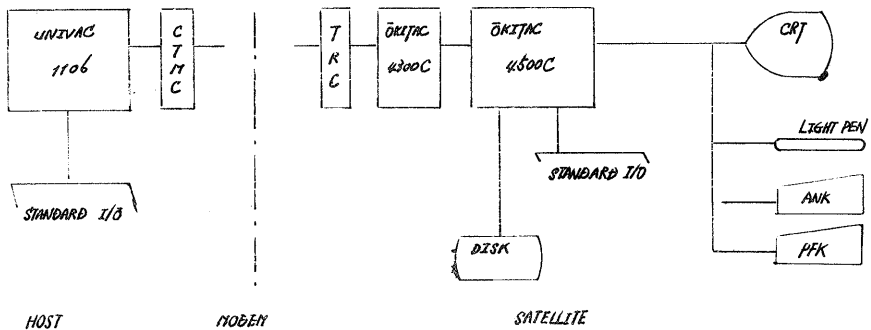


FIG.3. システム・ハードウェア構成

AGENT は、サテライトとして OKITAC-4500C、ホストとして UNIVAC-1106 を用いた 2 プロセッサ構成のグラフィック・システムである。

サテライトには回線制御用に小型コンピュータ OKITAC-4300C がおかれ、2400 BPS 以上の通信回線を通して、ホストとのデータ転送を行う。

AGENT システムでは、ホスト、サテライトの LOAD SHARE は FIG. 4 の様に行なわれる。



FIG. 4 ホスト、サテライトの LOAD SHARE

そして、分割されたグラフィック・プログラムおよび、システム間のデータ・リンクは双方におかれにデータ転送プログラム DTP/H、DTP/S によって行われる。ホスト、サテライトには各々に SHARE された LOAD にしむがって、グラフィック・ソフトウェアが、サブルーチン・パッケージ形式でおかれる。システム・ソフトウェア構成を FIG. 5 に示す。

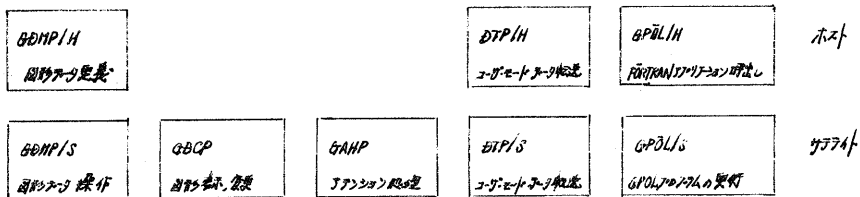


FIG. 5 システムソフトウェア構成

GDMP/S、および、GDMP/H

図形の階層性、図形相互間の論理的関係を表現するために、図形データの構造化が必要である。このためのソフトウェア・サポートとして、サテライト側にリンク構造のデータ・ストラクチャ・パッケージ GDMP/S がおかれる。GDMP/H は、ホスト側グラフィック・プログラムで図形データ構造と定義するのに用いられる。

GDMP/H を用いて、ホスト側で定義された図形データは、DTP/H によってサテライト側の図形データ・ファイルに自動的に転送される。

GDMP のデータ構造、および、構造化の例を FIG. 6 に示す。

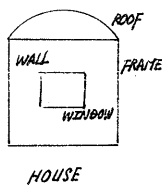


FIG. 6a

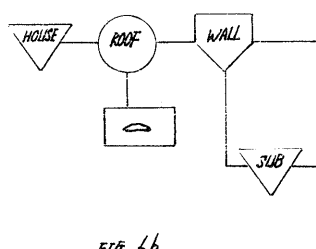
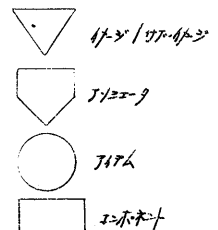


FIG. 6b



・ GDCCP

図形表示、表示図形の追加、修正、変更などの要求に反応して、図形データのディスプレイ・コマンド列への変換、ディスプレイ・コマンド列の修正追加などを行なう。

・ GAHP

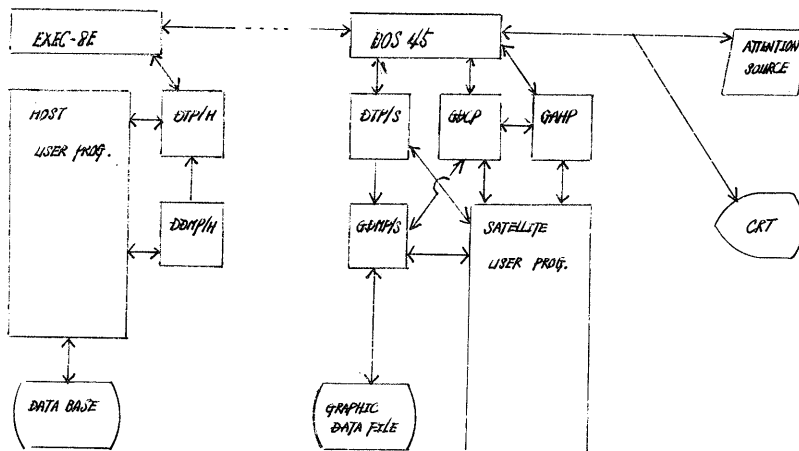
ディスプレイ・コンソールには、ライトペン、アルファニューメリック・キーボード、ファンクション・キー群から成る34個のアテンション・ソースがおかれているが、GAHPは、これらのアテンション・ソースのACTIVATION/INACTIVATIONや、アテンション情報の取出し、編集を行なうサテライト側のコア・レジダント・プログラムである。

・ DTP/H、DTP/S

ホスト、サテライトのグラフィック・プログラムあるいはグラフィック・システム間のデータ転送のために、ホストには、DTP/H、サテライトにはDTP/Sがおかれている。ホスト、サテライトのコミュニケーションは、DTP/H、DTP/Sを用いて、ユーザの管理のもとに行われる。

上に述べた各ソフトウェアは、全て、FORTRAN CALLABLEなサブルーチン・パッケージであり、反応して、これらを用いてFORTRANでグラフィック・プログラムを作成する事ができるが、特に、インタラクティブな処理の多いサテライト側プログラムの作成を容易に行えることが必要である。このために、サテライトユーザには、対話型グラフィック言語GPOLが提供される。

FIG.7に、AGENTシステム動作時のデータ・フローを示す。



→ 7-9の注を参照。

FIG.7 AGENT DATA FLOW

3.2. 特徴

AGENT システムは、グラフィカルな処理を全て、サテライトに SHARE することによって、

(1). コンソール・オペレーションによるホスト側システムのオーバ・ヘッドが消失する。

(2). オペレータ・アクションに対するレスポンスが速くなる。

という利点をもっている。

また、ホスト側グラフィック・プログラムは、デマンド・モードで動作し、サテライトからのメッセージ待ちの時は、外部記憶に SWAP されるため、

(3). ホスト側グラフィック・プログラムのコア占有率は小さい。

更に、JKITAC-4500C は中規模のデータ処理能力をもっているため、

(4). STAND ALONE モードで図形処理、あるいは他の業務処理を行う事ができる。

などの特徴がある。

3.3. 対話型言語 GPÖL

3.3.1. 目的

AGENT システムの各ソフトウェアは全て、FORTRAN CALLABLE なインタフェースを備えており、これを用いて FORTRAN IV レベルでグラフィック・プログラムを作成することができる。

AGENT では、ユーザ間によく知られてゐる事、グラフィック・プログラムの内、高度な数式処理を必要とする業務用アプリケーションの記述に有利である事から、ホスト側グラフィック・プログラムは FORTRAN で書かれる。

しかしながら、サテライト側グラフィック・プログラムの作成には、図形処理、特にオペレータとのインタラクションの記述に適したグラフィック言語が必要である。

GPÖL はノン・プログラマ・レベルのユーザが容易に図形処理を記述できる事を目的としてサテライトユーザに提供される専用型グラフィック言語である。

3.3.2. 特徴

設計プロセスが通常、反復試行をとともなう何段階かのパターン処理の積重ねからなっている事を認識するならば、グラフィック・システムには、これらのパターン処理機能の記述と、各処理単位間の実行スケジュールの記述の手段が提供されなくてはならない。

GPÖL では各処理単位にプログラム・ステートを対応させ、プロセスの進行をプログラム・ステートの遷移としてとらえ、ステート・ダイアグラム形式で実行スケジュールを記述する。

(1) DIAGRAM の記述

各パターン処理プログラムのスケジュールを記述するために DIAGRAM が用いられる。

FIG. 8 に示すプロセスは次の様に表わされる。

01 , DIAGRAM(NNN)
02 , LBTN(B1-B5)

```

03 STATE1, EX(MC, B1)
04 STATE2, EX(STATE1, B2)
05 STATE3, EX(STATE2, B3)
06 STATE4, EX(STATE2, B4)
07 MC, EX(STATE3, B5)
    , EX(STATE4, B5)
08 , END

```

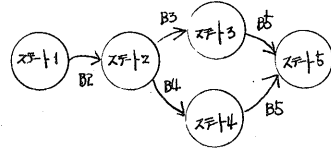


FIG. 8

01. 08 DIAGRAM の定義の開始、終了を各々示す。
02. ライトボタン B1-B5 を状態遷移のアテンションとして用いる事を示す。
この状態には、GPOL システムの動作モードが対応している。これを MC ステートと言う。
03. ライトボタン B1 アテンションによって、ステート 1 を移る事を示す。
STATE1 はステート 1 の名前である。
ライトボタン B2 アテンションによってステート 1 を脱し、ステート 2 に移る。
07. ライトボタン B5 アテンションによって、GPOL ステートへ移る事を示す。

(2) STATE の記述

DIAGRAM 中で定義された各ステートは実際には、更に幾つかのサブステートの遷移から成るステート・ダイアグラムで表わされる。

例えば、FIG. 8 中のステート 1 が FIG. 9 に示す構造をもつ時、ステート 1 は次の様に記述される。

```

08 , STATE(STATE1)
09 , FK(1-3)
10 SUBS1, EX(LMC, SUBSE, F1)
11 SUBS1, EX(SUBS1, F2)
12 LMC, EX(SUBS1, F3)
13 , END

```

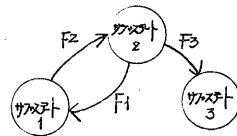


FIG. 9

08. 13 各々 STATE1 の定義の開始、終了を示す。
09. ファンクション・キー 1~3 をサブステート間の遷移アテンションとして用いる事を示す。
この状態には、ステート 1 上で GPOL システムの動作モードが対応している。これを LMC ステートと呼ぶ。
11. ファンクション・キー F2 アテンションによって、サブステート 2 へ移る事を示す。
SUBS1 はサブステート上の名前である。
ファンクション・キー F1 アテンションによって、サブステート 1 へ、ファンクション・キー F3 アテンションによって、LMC ステートへ移る。

DIAGRAM, および, STATE は, CRT上から直接, 定義する事もできる。
また, DIAGRAM は任意の時点で, STATE は, そのSTATE 中でいつでもオペレータによって CRT上に表示させる事ができる。

(3) SUBSTATE の記述

サブステートは, 次の2種のプログラムが対応してゐる。

- ・ GPHASE

処理すべきパターンの表示に必要な図形を定義するプログラムである。

- ・ OPHASE

GPHASE で作成された図形群の表示, 修正や, FORTRAN アプリケーションを用いた演算, 解析処理などからなるプログラムである。

(4) 図形の記述

GPOLでは, 図形は次の6種の基本図形と2種の図形演算子を用いた, 図形演算式によって記述される。

基本図形: POINT(点列), LINE(折線), SGM(T(線分群)

CIRCLE(円), ARC(円弧), TEXT(文字列)

図形演算子: '+' 結合を表わす

'=' 代入を表わす

図形演算式によって表わされた図形は PICTURE と呼ばれ, 図形操作の対象となる。

例えば, FIG. 6 に示す図形は, 次の様に書かれる。

```
ROOF(ATR1) = ARC(X1, Y1, X2, Y2, X3, Y3) + SGM(T(X2, Y2, X3, Y3, 1)
SUB          = WINDOW(X4, Y4) + FRAME(X5, Y5)
WINDOW(ATR2) = LINE(X6, Y6, 5)
FRAME(ATR3)  = LINE(X7, Y7, 5)
HOUSE = ROOF(X8, Y8) + WALL(SUB, X9, Y9)
```

ATR1, 2, 3 は各々, PICTURE ROOF, WINDOW, FRAME の表示属性を示すパラメータである。

また PICTURE SUB は, PICTURE WINDOW および FRAME から構成されており SUB 内での各々の位置は, X4, Y4, および X5, Y5 で表わされる。

この様に図形演算式を用いて, 図形の階層や, 構造はインプリシットに記述することになる。

PICTURE の内, GPHASE で定義されたものは, この GPHASE に対して, DISPLAY ステートメントが用いられない限り表示されないが, OPHASE では定義と共に表示される。

(5) アテンション処理の記述

GPOLでは先に述べた様に, プロセスについて, あるいは, その中のサブ・プロセスについてのスケジュールの記述は, ステート・ダイアグラム形式で行う。

ここでは、OFFPHASE中でのアテンションによる図形の操作、アプリケーションの実行などについて述べる。

一般に GPDL での記述形式は、

LA, DSC

の様になっており、アテンションLAの生起により、DSCで示された処理を実行する。

例えば、FIG.9で示した図形の内、PICTURE WINDOW がライトペンでピックアップされたら、これをN倍に拡大するためには

LP(WINDOW), WINDOW=SCALE(WINDOW,N)

と書けばよい。

SCALEの他に、ROTATE, MOVE, などの幾何学的な操作を表わすファンクション、OFF, ON, CHANGE などの表示属性の変更を表わすファンクション、

あるいは、図形演算式が許される。

また、

F1 BR(ANL, P1, P2, *)

は、ファンクション・キー F1アテンションによって、FORTRAN サブルーチン ANL を実行する事を示している。

P1, P2 は サブルーチン ANL の引数、* は、サブルーチン ANL がホストにある事を示している。

4. あとがき

AGENT システムでは、

- ・OPEN型構成を採用したことにより、ユーザ・モードのデータ転送の自由度が与えられるが、反面、プログラミングが難しくなる事。

また、GPÖLについていえば、

- ・インタラクティブ・プロセスのスケジューリング記述に主張をもっているが、FORTRANなどとホスト言語とするプリ・プロセッサ型言語のみの数式処理機能は殆んど有していない事。

など、問題点は点在する。

しかしながら、特に、インタラクティブ・システムでは、評価は、本来、使ってみて定まるものである。

AGENT システムは、近く、その上でグラフィック・アプリケーションが稼動する。

AGENT システムの諸機能、特に、そのインタラクション機能に関する問題点は、この時点で明らかになるう。

今後、この点について報告を行うつもりである。

参考文献

[1]. E.M. Thomas, "System Considerations for Graphic Data Processing",

COMPUTERS and AUTOMATION,

November, 1967 PP. 16 - 19