

構造化マルチウィンドウを使った CG統合環境:FUSION/MMI

林 一司, 村上公一
(株)富士通研究所

コンピュータ・グラフィクスにおける様々な処理を統合化するための枠組みとなるFUSION/MMIについて述べる。マウスや各種のウィンドウ・ツールを使用することにより、データや処理の流れが視覚化された、操作性のよい環境が構築される。

モデリングに要求されるような複雑な構造をもったデータを視覚的に扱うためには、ウィンドウ間に結合関係を設定できる構造化マルチウィンドウを提案する。結合関係の記述にはアイコンとスロットを用いた結合方式(アイコン・スロット方式)を用いる。

Structured Window System : FUSION/MMI

Kazushi Hayashi, Koichi Murakami

Computer-based Systems Laboratory, Fujitsu Laboratories LTD, Kawasaki
1015, Kamikodanaka, Nakahara-ku, Kawasaki, 211, Japan

In this paper, we will describe the FUSION/MMI, a framework to integrate various types of processing for computer graphics into multi-window environment. Full use of mouse and window tools, such as button, slider and menu, can make operations intuitive.

Since modeling in computer graphics requires complex data structure and manipulations, very flexible and powerful operations are needed. we propose a icon-slot method in which a child window is installed to the slot of the parent window, so relationship between windows is established. By mapping a window to a datum or manipulation, modeling operation can be done using the icon-slot method.

1. はじめに

計算機の利用分野が広がるに従って利用者も計算機の非専門家が多くなる。これはコンピュータ・グラフィクスの分野においても例外ではない。特にコンピュータ・グラフィクスでは、人間に馴染みやすい映像情報を出力するのに対して、モデリング、データ入力等の入力に対するインタフェースが貧弱である。このため、デザイナー等が手軽に利用することができない。これらの人にも容易に利用できるような使い易い計算機システムが重要となる。

このようなシステムを考える場合にはユーザ・インタフェースが非常に大きな意味をもつ。一つのアプローチとして、ユーザ・インタフェースに視覚情報を用いる方式がある。例えば、アイコンとよばれるシンボルを用いてコマンドの指示をさせるシステムや、さらに進めて直接これを用いてプログラミングを行うシステムがこれに相当する。

Show & Tell に代表されるようなシステム [1, 2] では、一つの処理を一つのアイコンで表し、ユーザはそれらのアイコンの間をデータフローを表す線で結ぶことにより処理を進めてゆく。この方法は個々の処理が比較的簡単で、データがフラットな場合にはかなり有効であると思われる。しかし、コンピュータ・グラフィクスにおけるモデル構造の記述は複雑であるので、モデリング処理をアイコンとデータフローを表す線分だけで扱うのは困難である。

良好なユーザ・インタフェースを実現する為には多くの処理を組合せなければならない。しかし、これらの処理を巨大な一つのプロセスで実現することは望ましくない。そこで処理をいくつかの機能的に独立した部分に分割し、各々の処理を別々のプロセスで実行させる（例えばAdagio [3]）ことが考えられる。このようにすると、個々のプロセスはアクティブなオブジェクトとして考えることができ、オブジェクト指向のパラダイムが利用できる。この形で構成されたシステムは、機能の追加・変更等に柔軟に対応することができ、保守も容易である。

我々は分散プロセスをまとめる枠組みとしてマルチウィンドウに注目した。マルチウィンドウは複数の端末を表示するだけのものではなく、複数のウィンドウを組み合わせて使うことにより、もっと自由にユーザ・インタフェースを構築することができるものである。アプリケーションが複数のアイコンやウィンドウを関係ずけて用いることにより、より優れたユーザ・インタフェースを実現することができる。

我々はマルチウィンドウに、データ構造を自然に記述できるようにウィンドウ間に構造記述を導入した構造化ウィンドウを考案した。このマルチウィンドウ上では処理とデータ構造の表現・操作を統一的に扱うことが可能であり、単なるカットアンドペーストでは不可能なウィンドウ間の関係が達成できた。これらの機能に基づいてコンピュータ・グラフィクス用のUIMSとしてFUSION/MMIを開発した。

2. 構造化ウィンドウ

ウィンドウ構造の記述方法としては、リンク関係をアーク（線分）を用いて表現する方法もあるが、複雑な構造に対しては線分が入り組んでしまい分かりづらくなる。そこで、アイコンとスロットをもちいた結合方式を（アイコン・スロット方式）を開発した。

本方式では、各ウィンドウは自分の領域内に結合関係を示すための領域を持ち（スロット）、ユーザが結合するウィンドウを指定すると、そのウィンドウがアイコンに変換されてこのスロットに表示される。この操作を繰り返すことにより、ウィンドウ間の結合関係を構築してゆく。ウィンドウとアイコンにはそれぞれ他のウィンドウまたはアイコンと結合された状態（Linked）とされていない状態（Free）の二つの状態があり、結合操作によってこれらの状態間を遷移する（図1）。結合操作例を図2に示す。ウィンドウw1とウィンドウw2を結合するときには、まずw2をアイコンi2にし、これをウィンドウw1のスロットに移動・結合する。ウィンドウのスロットにアイコンが入ると、アイコンの輝度が反転して結合されたことを示すようになる（Linked Icon）。同時にi2は画面から消える。結合した状態のままデータ操作を行う為に、w2をi2としないでw1に結合することも可能である（Linked Window）。

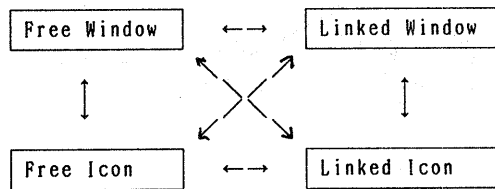


図1 ウィンドウの状態遷移

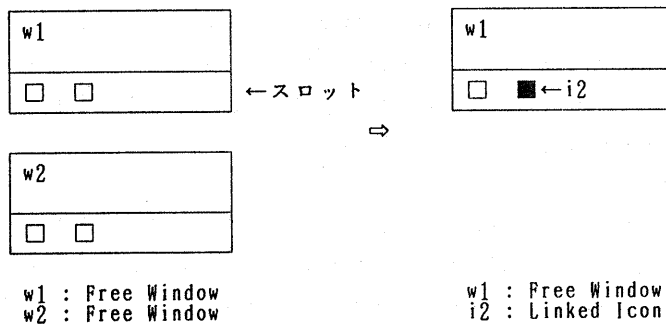


図2 ウィンドウの結合の設定

本方式では、複数のウィンドウ間の結合関係を自由に設定できるので、Tree構造やNetwork構造をもつデータを自然に記述することができる。

3. データベース

FUSION/MMIでは、ユーザが記述したデータや処理の結果はウィンドウに割り当てられているプロセスが持っている。これらの情報は個々の処理に関する断片的なものであり、システム中で1つにまとめなければならない。そこで、これらのデータを管理

するためのデータベースを設けた。各プロセスはシステムが必要とする情報をメッセージとしてこのデータベースに送信する。データベースは送信されてきたメッセージに従って処理を行う。ウィンドウ間に新しい結合関係が設定された場合は、データベース内でもそのウィンドウに対応しているデータ間で結合関係が構築され、データが変更された場合は、データベース内のデータもそれを反映するように変更される。これによりウィンドウ毎に異なる各々の処理がデータベースを介在として結合される。

4. FUSION/MMIの実現

4.1 構成

FUSION/MMIは全体として三種類のプロセス群から構成されている(図3)。現在のシステムは、SUN ワークステーション上でSunView を用いてインプリメントされている。

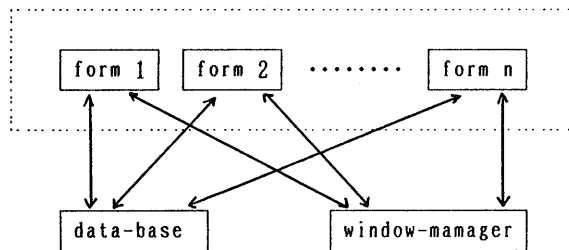


図3 プロセスの関係

(1) フォーム

FUSION/MMIではウィンドウとそれに対応しているプロセスを合せてフォームと呼ぶ。ユーザはこのフォームを通して処理を行う。各フォームはユーザが自然な感じで扱えるようにデザインされている。操作は、メニュー・ボタン・スライダ等のウィンドウ・ツールを多用し、マウスによる操作を中心とする視覚的な環境下で行えるようになっている(図4)。

(2) データベース

データベースは各フォームから送信されてきたデータを一括管理している。フォームの生成・消滅・更新・階層化に対応したデータの生成・削除・更新・階層化を行う。フォームとデータベースの間は、プロセス間通信によって結合されている。データベースは構造データを管理しなければならない。また複雑なデータ操作に柔軟に対応しなければならない。そこで内部表現としてはフレーム構造を採用し、データベースをlispのプロセスとして記述してある。個々のフレームが一つのフォームに対応することになる。

(3) ウィンドウマネージャ

このプロセスは、システムでのマルチウィンドウ全体の制御を行っている。ユーザからのフォームの生成・消去の要求、フォームやアイコンの移動、入出力の管理等を行っている。

4.2 処理

FUSION/MMIでは、通信はデータベース・プロセス \leftrightarrow フォーム・プロセス、ウィンドウ・マネージャプロセス \leftrightarrow フォーム・プロセス間で行われる。プロセスは、事象が発生するとその事象を表すメッセージとデータを指定されたプロセスに送信する。このメッセージを受信したプロセスは、そのメッセージを解析して対応した処理を行う。階層関係で上位に位置するフォームをA、下位に位置するフォームをBとしたときに、階層化の手順は以下ようになる(図5)。

- ① フォームBの内部データを保存する為に、Bはデータベースにデータを送信する。
- ② フォームA上で、Bの入る(表示)されるスロットが指定され、そのスロット番号がデータベースに送られる。
- ③ データベース内で、フォームAのデータとフォームBのデータ間の階層化が行われる。
- ④ データベースプロセスが、フォームAに対してそのスロットに表示するアイコンの情報をおくる。
- ⑤ フォームAは指定されたスロットにフォームBのアイコンを表示する。
- ⑥ データベースプロセスはフォームBに対して終了メッセージをおくる。
- ⑦ フォームBは自分のプロセスを終了する。

5. まとめ

データ構造を自然に記述できるようにウィンドウ間に構造記述を導入した構造化ウィンドウを考案し、これを用いて構造をもつデータの操作を視覚的な環境下で行うことを可能にしたマルチウィンドウ環境を構築した。そして、これに基づいてコンピュータ・グラフィクス用のUIMSとしてFUSION/MMIを開発した。現在、FUSION/MMIを用いて絵言語モデラFUSION/MODELERを開発中である[4]

本システムの機能は、コンピュータ・グラフィクス以外でも、構造をもつデータを扱う分野(たとえば、CAD等の設計データあるいはエキスパートシステムなどで用いられるフレーム型の知識の操作)等でも有用であると思われるので、今後はそのような分野への適用も行って行きたいと考えている。

[謝辞] 本研究に対して議論等で御援助下さいました石井システム研究部長と白石研究室長に感謝します。

[参考文献]

- [1] 今井: "ビジュアル・プログラミング言語show and tell", 第161回情報科学研究会資料(昭和61年7月8日), 慶応義塾大学情報科学研究所
- [2] 岩田: "アイコンプログラムの作成・実行環境の開発", 61年情処全国大会後期
- [3] Peter P. Tanner et al.: "A Multitasking Switchboard Approach to User Interface Management", Computer Graphics 20,4(August 1986), 241-248.

[4] 村上：”（言語+対話）型CGモデラ：FUSION/MODELER”，集中研究会資料，1987年 8月20～21日

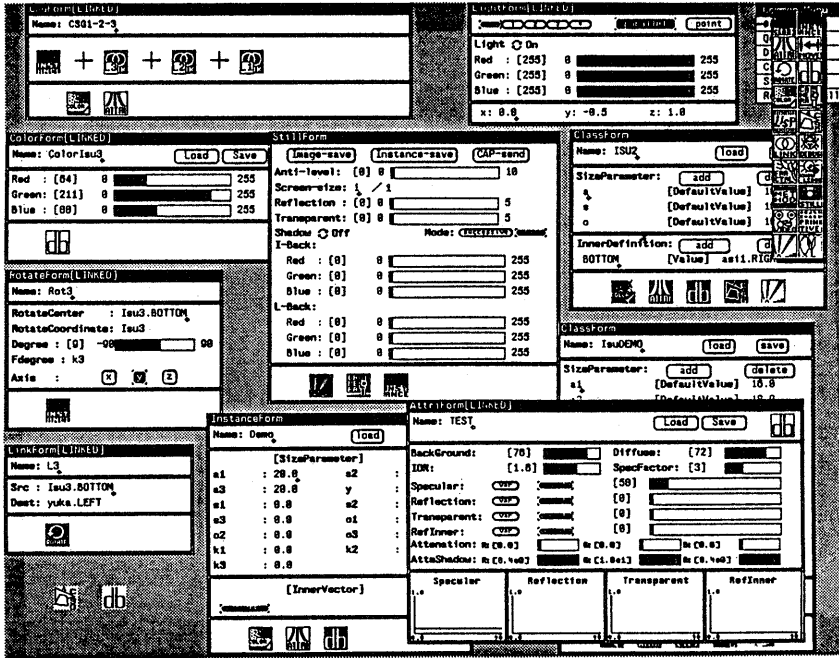


図4 各種のフォーム例

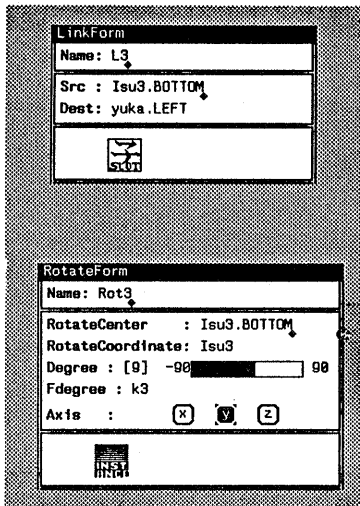


図5-1 結合前

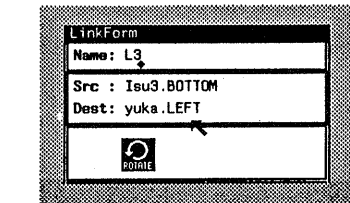


図5-2(a) 結合後(Linked Icon)

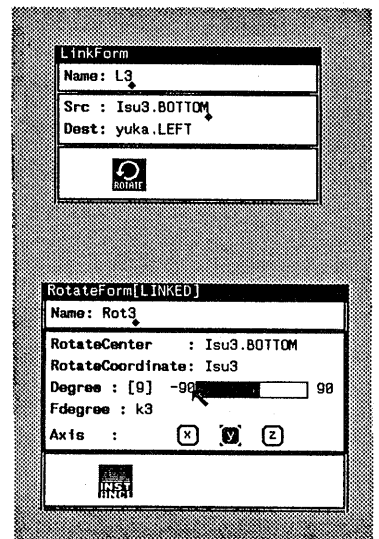


図5-2(b) 結合後(Linked Window)

図5 フォームの結合