

3G-5

並列オブジェクト指向言語 Fleng++ によるユーザインタフェースの構築*

勝亦 章善[†] 小池 汎平[‡] 田中英彦[‡]富士通株式会社[†] 東京大学工学部[‡]

1 はじめに

現在、誰もが容易にウィンドウ・システムを利用できる状況になり、より高度なグラフィカル・ユーザインタフェースの要求が高まっている。しかし、ユーザインタフェースが以前と比較ならないほど表現能力及び操作性が向上した反面、ウィンドウ処理はより複雑化し、プログラム開発の妨げになっている。今までは特定のプログラマだけが、この負担を負っていたが、グラフィカル・ユーザインタフェースが身近になった現在、ユーザインタフェースを容易に作成できる環境を、いかにして提供できるかが問題となっている。

今回、我々はより柔軟で使い易いユーザインタフェースを実現するため、本研究室で開発した 並列オブジェクト指向言語 Fleng++ [1, 2] を用いて、ユーザインタフェースの構築を行った。Fleng++ は、本研究室で開発中の高並列推論エンジン PIE[3] 用に開発された言語であるが、PIE は現在開発中のため、SUN 上に作成したインタプリタに対し、X-window システムをベースに実装を行った。

本稿では、ユーザインタフェースの概要について述べる。

2 Fleng++ によるユーザインタフェース

従来の X-window システムをベースにしたプログラムやツールキット (X-Toolkit [4], CLUE [5] など) では、イベント起動モデルを採用しているため、固有イベント処理をコールバック関数として設定し、メインループに制御を渡すことによって処理が行われている。この場合、主体はメインループであり、そのメインループから呼ばれる関数を作成することになる。このような受身的なプログラミング・スタイルは、慣れていない場合、作成し易いものとはいえない。

このような方式に対して、Fleng++ ではイベントをメソッド・コールに変換することにより、オブジェクト指向プログラミングにおける自然な処理の流れとしてイベント処理を組み込む方法を採用した。これにより、プログラミング・スタイルを変更することなしに、ウィンドウ環境を扱うことが可能になる。

Fleng++ は Committed-Choice 型言語 Fleng に対し、オブジェクト指向機能を導入した言語である。そのため、ユーザインタフェースを構築する際に以下の利点を持つ。

- 論理言語の柔軟性
- Committed-Choice 型言語の入出力処理の扱い易さ
- 並列パラダイムによる並列処理の記述能力
- オブジェクト指向機能によるシステム記述能力

3 ユーザインタフェースの階層構造

Fleng++ のユーザインタフェースは、基本的に 図1 に示すように、インタフェース層、ツールキット層、アプリケーション

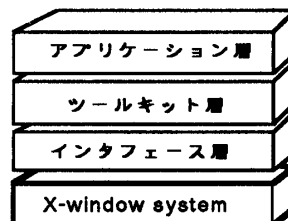


図1: ユーザインタフェースの階層構造

層の三階層から構成されている。

3.1 インタフェース層

インタフェース層は、ウィンドウ・システム固有の処理を吸収し、抽象化した機能を上位層に提供する層である。ウィンドウ・システムの資源 (GC、ピクスマップなど) を操作する専用クラスを設け、そのクラスを使用して資源管理を行う。最も基本的なクラスを以下に示す。

- xwindow クラス
ウィンドウ・システムと直接インタフェースを持つクラスである。ウィンドウ・システム資源を扱うためには、このクラスを継承する必要がある。
- event クラス
イベント関連の処理を扱うクラスである。処理系からイベント情報をイベント・ストリームとして入力し、メソッドコールへの変換を行う。
- window クラス
狭義のウィンドウ処理を行なうクラスである。ウィンドウの生成、消去などを扱う。

3.2 ツールキット層

ツールキット層は、インタフェース層から提供される機能を用いて、ユーザインタフェースを構築するための基本的な骨組みを形成する層である。ツールキットは、図2に示すような基本処理部、補助操作部、部品から構成されている。

3.2.1 基本処理部

基本的な処理方式として Smalltalk-80 で有名な MVC 構成法 [6] を採用している。model, view, controller の抽象化クラスを設け、これらのクラスを継承することにより MVC を実現している。MVC は 実際のウィンドウに対応し、controller

* "Construction of User Interface in Parallel Object-Oriented Programming Language Fleng++"

[†]Akiyoshi KATSUMATA, FUJITSU LIMITED

[‡]Hanpei KOIKE, Hidehiko TANAKA, Univ. of Tokyo, Faculty of Engineering

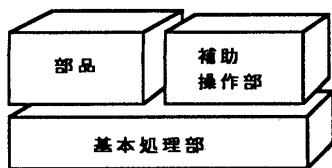


図2: ツールキット層の階層構造

は event クラスを継承してイベント処理を受け持ち、view は window クラスを継承してウィンドウ上の表示の管理を行う。

また、ウィンドウ単位のイベント・ストリームを発生させることにより、他のウィンドウとの独立性の高いプログラムが作成でき、ウィンドウの並列処理を行うことができる。

3.2.2 補助操作部

ウィンドウ枠の設定や、メニュー操作など基本処理部に対する補助的な操作を行う。基本処理部の一部として動作するが、拡張性を考慮して別パートとした。

3.2.3 部品

ツールキットにおける部品は、VC 組で作成される。この部品をモデル (M) と結合することによりプログラムを作成することができる。プラグマビュー (pluggable view) を採用することにより、同様な機能を持つ部品の再利用を向上させ、効率の良いプログラム開発を可能にする。標準ビューとして codeView, listView, formView を提供している。

3.3 アプリケーション層

アプリケーション層は、ツールキット層の機能を使用して応用プログラムを作成する層である。プログラミング環境のために開発されるツールはこの層に属する。

4 ユーザインタフェースの実装

ウィンドウ処理の中心であるイベント処理を扱う event クラスの実現方法について述べる。

event クラスは、初期化メソッド (init) でイベント・ストリームを入力し、そのイベント・ストリームを内部のローカル述語 (eventInput) によりイベントを入力し、メソッド・コールに変換する。

4.1 イベントの取り込み

ローカル述語 (eventInput) は、イベント・ストリームが終了するまで再帰呼び出しでループする。取り込んだイベント情報は、第2引数に持ち回っているメッセージ・ストリームに、:event(S,Event) の形式で束縛され、メソッド・コールに変換される。

```
eventInput([], !close(_)).
eventInput([Event|L], !{[event(_, Event)], S}) :-
    eventInput(L, S).
```

4.2 イベント処理メソッドの定義

各イベントの処理は、以下の形式で定義されたメソッド定義で行われる。event クラスでは、上位クラスの受け皿として全てのイベント・タイプを定義している。

```
:event(S, expose(ID, X, Y, W, H)).
:event(S, buttonPress(ID, X, Y, N)).
```

...

5 プログラミング環境

ユーザインタフェースをベースに、各種のツールを作成することにより、Fleng++ のプログラミング環境の充実を図ることができる。

5.1 ブラウザ

Smalltalk-80 ライクなブラウザである。プログラムの作成、定義述語の検索、定義節の動的な変更などの処理を行う。

5.2 グラフィカル・エディタ

対話型プログラムの開発支援を行なうツールである。画面上に表示されている部品をインタラクティブに組み合わせて、表示イメージを基にプログラムを作成するエディタである。この様なツールとしては、NeXT の Interface Builder や SUN の GUIDE などがある。グラフィカル・エディタの特徴は、動的リンク機能により、作成と同時に実行が可能な点である。これにより開発サイクルの短縮を図ることができる。

6 おわりに

Fleng++ の柔軟性及びモジュール性は、柔軟なユーザインタフェースを効率良く構築する際に非常に有効であった。

今後は、ツールの開発などプログラミング環境を充実させてゆく予定である。

参考文献

- [1] 中村宏明, 田中英彦, “並列オブジェクト指向言語 FLENG++ の実装”, 第38回情報処理学会全国大会 6Q4(1989)
- [2] 中村宏明 “Committed-Choice 型言語に基づいたオブジェクト指向プログラミング・システムの研究”, 東京大学工学部情報工学専攻修士論文 (1989)
- [3] Hanpei Koike and Hidehiko Tanaka, “Multi-Context Processing and Data Balancing Mechanism of the Parallel Inference Machine PIE64”, Proc. of Int. Conf. on FGCS 88', pp.970 - 977.
- [4] “X ToolKit Intrinsics - C Language Interface, X Window System, X Version 11, Release 3”, 1989.
- [5] “Common Lisp User Interface Environment”, 1988.
- [6] 上谷晃彦, “統合化プログラミング環境 Smalltalk-80 と Interlisp-D”, 丸善.