

# オブジェクト指向分散環境OZ++のGUI指向アプリケーション開発ツール: OZ++/Frame

3H-3

中村章人<sup>†</sup> 中川 祐<sup>\*‡</sup> 塚本享治<sup>†</sup>

<sup>†</sup> 電子技術総合研究所    <sup>‡</sup> 富士ゼロックス情報システム

\* 開放型基盤ソフトウェアつくば研究室研究員

## 1 はじめに

OZ++/Frameは、GUIを利用したアプリケーション開発ツールであり、プログラマ向けのGUIツールキットであるコアシステムライブラリ、GUIを表示するためのグラフィックエージェント、オブジェクトの属性を変更するためのミューテータから構成される。OZ++/Frameの目的は、(1) プログラマにGUIライブラリを提供する、(2) エンドユーザであってもGUI上でコンポーネントの組合せや編集を行いながら目的のアプリケーションを開発できるようにする、(3) GUIを含むコンポーネントをネットワーク上で容易に共有・再利用できるようにすることである。

このような目的のツールは多数あるが、OZ++/Frameの特徴は、アプリケーションおよびコンポーネントをネットワークワイドに公開し再利用できる仕組みを提供していることである。これは、広域ネットワーク上でオブジェクトを転送し、必要なクラスを自動的に配送するというOZ++の機能[1]を利用したものである。

OZ++/Frameで作成したアプリケーションおよびコンポーネントの共有と再利用は、ジャンクショップというネットワークワイドにアクセス可能なオブジェクトを利用して行う。OZ++/Frameで開発したコンポーネントをジャンクショップに公開し、またジャンクショップにあるコンポーネントをOZ++/Frameに取り込むことができる。ジャンクショップおよびOZ++/Frameは永続オブジェクトなので、アプリケーションの開発および実行をオブジェクト(インスタンス)の状態から開始できる。本論文では、OZ++/Frameおよびジャンクショップの機能と特徴について述べる。

## 2 OZ++/Frame

本章では、OZ++/Frameの構成と機能について述べる。

A GUI-based development tool OZ++/Frame in OZ++: an object-oriented distributed environment

Akihito Nakamura (Electrotechnical Laboratory),  
Yu Nakagawa\* (Fuji Xerox Information Systems, Co., Ltd.),  
and Michiharu Tsukamoto (Electrotechnical Laboratory)

\* Researcher, Tsukuba Laboratory, Open Fundamental Software Technology Project

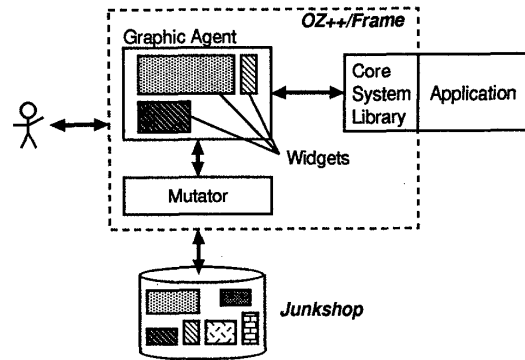


図 1: OZ++/Frameの構成

### 2.1 コアシステムライブラリ

コアシステムライブラリは、OZ++プログラマ向けのGUIツールキットである。ボタンやテキストフィールドなどの一般的なGUIウィジェットに加えて、スクリーンとスライドという特殊なウィジェットを提供している。コアシステムライブラリのインタフェースは、プログラマに対してGUIを利用するための枠組を提供する。

#### 2.1.1 ウィジェット

スクリーンとスライドは、HyperCardで用いられているカードとバックグラウンドとほぼ同じ機能を提供する。すなわち、一つのスクリーンに対して複数のスライドが存在し、一度に表示されるのはスクリーン一つとスライド一つである。スライドはスクリーンにスーパーインポーズされて表示される。スクリーンは、ウィジェットを保持し、それぞれの表示位置を管理する。スライドは、各ウィジェットのデータ値を保持する。例えばテキストフィールドでは、その表示位置はスクリーンが保持し、表示する文字列はスライドが保持する。

#### 2.1.2 アプリケーションとの結合

GUIとアプリケーションを結びつけるには、モデルという抽象クラスを利用する。各アプリケーションでは、モデルの具象クラスを用意する。

各ウィジェットにはモデルのインスタンスを結合することができ、複数のウィジェットに一つのモデルを結合することもできる。グラフィックエージェントがユーザ

のGUI操作(イベント)を検知すると、OZ++/Frameが以下の処理を行う。

- (1) 操作したウィジェットに結合されているモデルを探す。なければ、ウィジェットを保持しているスクリーンのモデルを探す。
- (2) モデルが見つければ、ウィジェットの種類とイベントの種類とに基づいて、そのモデルの特定のメソッドを起動する。

(2)のメソッド起動の引数には操作されたウィジェットが渡されるので、モデルは対象のウィジェットを判別できる。プログラマは、このモデルのメソッドの中からアプリケーションに対して必要なメソッド起動を行うようにプログラムしておく。

## 2.2 グラフィックエージェント

グラフィックエージェントは、ウィジェットの定義と実際の表示を行う。プログラマに対してはコアシステムライブラリのインタフェースだけを提供し、グラフィックエージェントの存在は隠蔽している。これによって、実装者はウィンドウシステムや実装方法を選択できる。

グラフィックエージェントの現在の実装では、Tcl/Tkを利用している。Tclのインタプリタとスクリプトから構成されており、Tkで提供されているウィジェットをそのまま利用している。

## 2.3 ミューテータ

ミューテータは、エンドユーザ向けの開発ツールである。エンドユーザにとってコアシステムライブラリを用いたプログラミングは困難であるが、簡単なGUI操作ならばできる。そこで、既存のオブジェクトの修正や組合せをGUI操作のみで行えるようにし、これによって新たなアプリケーションを開発できるようにした。

オブジェクトに対するこれらの操作は、すべてインスタンス変数の変更のマッピングされる。これによって、ウィジェットのビュー(大きさや色など)はもちろん、結合するモデルも変更することができる。

## 3 ジャンクショップ

OZ++/Frameで作成したオブジェクトを、クラスと同様にユーザ間で共有・再利用できると便利である。ジャンクショップというネットワークワイドにアクセス可能なオブジェクトによって、このサービスを実現した。

ジャンクショップは、オブジェクトに名前を付けて登録できるリポジトリである。OZ++/Frameは、オブジェクトをジャンクショップに登録し、また登録されている

オブジェクトを取り出すことができる。オブジェクトは、単一のクラスのインスタンスでもよいし、複数のインスタンスから構成されるもの(例えばウィジェットを複数持ったスライド)であってもよい。オブジェクトがネットワークを通じて転送された場合でも、オブジェクト間の参照関係はOZ++システムが保証する[2]。

ミューテータとジャンクショップはプログラミングを行わないエンドユーザだけでなく、コアシステムライブラリを使うプログラマにとっても有用である。複数のウィジェットのビューをプログラムで調整するのは、難しくはないが面倒な作業である。必要なパラメータをすべてプログラム中に記述し、変更する度にコンパイルが必要である。例えば、ウィジェットを適切な位置に配置したり、色を調整したりするのは、GUIを用いて行う方が簡単である。オブジェクトのインスタンスを生成する代わりに、ジャンクショップからオブジェクトを取り出すことにすればこのような作業はGUIでできる。言い替えると、OZ++/Frameをインタフェースビルダとして利用することもできる。

## 4 まとめ

本論文では、OZ++のGUI指向開発ツールOZ++/Frameについて述べた。OZ++/Frameの利点は以下の通りである。

- OZ++/Frameは、GUIを含むアプリケーションの開発を、プログラマとエンドユーザの二つのレベルで支援することが可能である。
- ジャンクショップを利用することによって、開発したコンポーネントをネットワーク上で公開し、再利用することができる。

OZ++/Frameを利用して分散スプレッドシートを開発した。スプレッドシートでは、シートにあらかじめ実際のデータや計算式を入れたものなど、インスタンスを共有したい場合が多い。例えば、必要なデータおよび計算式を入れたシートをジャンクショップで公開し、データや計算式の更新を行いながらそれを共有できる。

この研究は情報処理振興事業協会(IPA)が実施している「開放型基盤ソフトウェア研究開発評価事業」の一環として行われたものである。

## 参考文献

- [1] 西岡他:オブジェクト指向分散環境OZ++システム第一版の実現, Aug. 1995.
- [2] Hamazaki, Y. et. al.: The Object Communication Mechanisms of OZ++, Proc. IEEE ICOIN-9, Dec. 1994.