

言語非依存型アプリケーション開発支援環境の試作

D-3-2

iML: Rapid Application Development System

水谷 伸晃 福田直樹 新谷虎松

Mizutani Nobuaki Fukuta Naoki Shintani Toramatsu

名古屋工業大学 知能情報システム学科

Dept. of Intelligence and Computer Science, Nagoya Institute of Technology

1. はじめに

近年では、GUIを持つアプリケーションが主流となり、各種アプリケーションを直感的に操作できるようになった。しかし、それに伴いアプリケーション全体のコード量を占めるユーザーインターフェース部分の割合は増大してきている。現在のアプリケーション開発支援環境はさまざま、プログラミング言語によっては、提供されていない場合も多くある。このような統一性のなさからくる問題点として、実装者に対して数多くのプログラミング言語・開発環境使用法などの習得など様々な面で負担が大きくなっている。共通した操作性を持ちユーザーが使用言語を選択することが可能なGUI構築ツールの必要性が上げられる。

UIMS(User Interface Management System)の研究などにより応用プログラムとインターフェースを独立に開発し、最終的に結合するということが可能になった。UIMSとは、文字通り機能部分と人の間にあり、相互間の不整合の最小化、生産性、満足度などの最大化が目的とされている。UIMS自身は、応用プログラムとGUI部分を効率よく構築するための実行ライブラリとその構築を支援するツールから成っている。

本研究では、アプリケーションの機能に対して適切な言語を選択でき、その上で共通した操作性をもつGUI構築支援システムとしてiMLを試作した。

2. User Interface Management System(UIMS)

ユーザーインターフェースは、複数の応用プログラムにおいて類似した構造を持つ。応用プログラムからユーザーインターフェースを分離し、インターフェース部分をモジュール化することにより、有用な共有ツールを作成することが可能である。分離の種類には論理的分離と物理的分離がある。論理的分離とは、ユーザーインターフェースとアプリケーションルーチンがデータを共有せず、限定した相互間の通信機能を有する場合である。物理的分離とは、ユーザーインターフェース記述がアプリケーションルーチンとは別に記述されている場合である。アプリケーションの開発における構造として適切な組み合わせは、論理的分離のない、物理的分離のあるシステムであると考えられる。

3. 本システムについて

本システムでは、UIMSにおける分離として論理的分離のない物理的分離をしたシステムを実現している。特徴として、適切な言語の選択とユーザーの負担軽減という観点から、アプリケーションルーチンを記述する言語をアプリケーション作成者が選択/追加できるという点が上げられる。

言語が外部からなんらかの方法により呼び出す手段を用意していることが前提となっているが、その条件を満たすならば、本システムが定める仕様を満たすJavaのinterfaceを実装したクラスを生成することにより使用言語を拡張する

ことが可能となっている。ユーザーは自分が使用したい言語における拡張を施すことにより本システムが対応するアプリケーションルーチン記述言語の選択肢の幅を広げることが可能なのである。

本システムの実行例を図1に示す。システムの構成はメニュー(図1-①)、モード設定ウィンドウ(図1-②)、プロパティ/イベント設定ウィンドウ(図1-③)、スクリプト編集ウィンドウ(図1-④)、そして実際のGUI設計ウィンドウなど(図1-⑤)となる。

現在メニューから選択可能なコンポーネントは、Java言語におけるAWTのパッケージが中心となっている。編集対象となるコンポーネントの追加も容易な設計をしている。ユーザー独自のコンポーネントを追加することによりシステムを拡張することが可能である。実際の処理の記述は、イベント設定ウィンドウからイベントを選択することにより、スクリプト編集ウィンドウに表示されたひな形を元にユーザーが具体的な処理を記述する形で行われる。コンポーネントのサイズ変更、追加/削除といったGUIの動的変更を行う処理においては、スクリプト上でJavaプログラムを単純な文字列として記述することで実現できる機能を提供する。

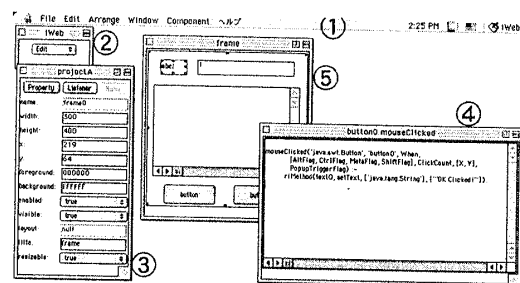


図1: アプリケーション開発環境の実行例

4. まとめ

本稿では、アプリケーション開発におけるGUI設計支援を行う機構の試作を行った。アプリケーション開発者が、その機能を実現するのに適している言語を選択し、かつ共通した操作性を持つ本システムを用いることによりGUIの構築が容易となったと考えられる。現在、選択可能なアプリケーションルーチン記述言語は、本研究室で開発された論理型言語MiLog [1]がある。今後の課題としては、各種言語の対応とRADツールとしてのインターフェース構築支援部分の機能の更なる拡張が上げられる。

参考文献

- [1] 福田直樹, 新谷虎松: "情報エージェント記述言語 Weblog におけるモビリティの実現", 第60回情報処理学会全国大会論文集 (3), pp.79-80, (2000.3)