

アプリケーション組み込み型の GUIプログラミング環境 Partskit

成田 真一† 大鎌 広‡ 藤原 祥隆†

北見工業大学 情報システム工学科† 室蘭工業大学 電気電子工学科‡

1 はじめに

従来 GUI プログラムはイベントの集中受け付けとコールバック関数に基づいたフレームワークで開発されてきた。この場合外観などの設計を行ってからコールバック関数を作成するが、コールバック関数の作成後に外観の設計変更があると、修正の負荷が大きい。

これに対し、GUI の各部分を独立して作成し、それらを組み合わせたプロトタイプに機能を追加していく GUI プログラミング方式 Partskit[1] を提案している。

本稿では、Partskit アプリケーションに組み込む GUI プログラミング環境について述べる。

2 Partskit

Partskit では、大きなアプリケーションを“部品”と名付けられた、単一の機能を持ち、各々が独立に疑似並行動作するクラスオブジェクトを、通信線で接続し組み合わせることで構築する。このため、部品は再利用可能かつ拡張可能な性質を併せ持っている。

Partskit が提供する機能の一つに、プログラムの設計を部品毎に独立して行えるという点がある。これらの再利用可能な部品をグラフィカルに管理することで、ユーザが全情報を一覧でき、アプリケーション構築が容易に行える環境 (アプリケーションエディタ) を開発している。

3 アプリケーションエディタ

3.1 概要

Partskit では汎用の部品が用意されている。それらだけで必要な機能が満たせなければ、既存の部品から継承して新たな部品を作成できる。アプリケーションエディタは、これらの部品の接続を支援するものである。

アプリケーションエディタを使用する事で、以下のような利点が得られる。

- 使用可能な部品の情報を明示し、必要な部品が見つけやすい。
- 部品の接続を GUI 上で行うので、プログラムの構成やデータの流れが把握しやすい。
- ソースコードを書かずに、容易にアプリケーションを作成できる。

このアプリケーションエディタをアプリケーションから随時呼び出せるようにする事で、既存部品から土台となる小規模なアプリケーションを作成し、それを元に徐々にプログラムを成長させていくといった開発方法を可能にする。

3.2 部品の動的ロード

Partskit で使用する部品を静的なオブジェクトファイルとして実現した場合、Partskit で作成したアプリケーションは、部品の仕様に変更があったり、部品の追加・削除、または接続状況の変化があれば、再度コンパイルしなくてはならない。アプリケーションエディタ自身も Partskit で作成されているため、部品を追加登録した場合などにはコンパイルし直す必要がある。これは、決まった部品を繋ぎ合わせるだけならさほど問題ではないが、新規の部品や既存のものを改良した部品を次々と追加していくとなると大きな足枷となる。

この事から、部品を動的にロードするシステムが開発された。[2] このシステムでは部品を共有オブジェクトファイルとし、使用する部品やそれらの接続情報を独自の起動スクリプトとして記述し、それをブートローダが読み込んでアプリケーションが実行される。

アプリケーションエディタは部品の共有オブジェクトファイルから、名前、使用方法、入出力ポートの型と数といった情報を取得し、ユーザに提供する。そして作成された結果を起動スクリプトとして出力する。

また、既存の部品が更新された場合でも、アプリケーションは当然再コンパイルする必要は無く、起動スクリプトを編集する必要もない。

Partskit GUI programming framework built into application
†Masakazu NARITA, Yoshitaka FUJIWARA
Dept. of Computer Sciences Kitami Institute of Technology
165, Koen, Kitami, Hokkaido 090-8507, Japan
‡Hiroshi OHKAMA
Dept. of Electrical and Electronic Eng. Muroran Institute of Technology
27-1, Mizumoto, Muroran, Hokkaido 050-8585, Japan

3.3 構成

アプリケーションエディタでは、データ部分とユーザインターフェース部分が完全に分割されている。これはデータの確一性の確保、機能限定による保守の容易さ、機能拡張(部品の追加)のためである。データを扱う部品は、ウィンドウを持たず表に現れない不可視部品として作成される。ユーザインターフェースとなる部品は、ウィンドウを持ち様々なイベントを受け取ることのできる可視部品で作成される。

アプリケーションエディタは、大まかに以下の4部品で構成される。

- 不可視部品 PartsList
使用可能な部品の情報を共有オブジェクトより取得し保持する。他の部品からの要求に応じてその情報を伝える。
- 不可視部品 ObjectList
現在作成中のアプリケーションを構成するオブジェクトと、それらの接続関係の情報を保持する。
- 可視部品 PartsPalette
部品をアイコンで表示して、どれを使うかをユーザに選択させる。選択された部品を、他の部品に伝える。
- 可視部品 Connect
PartsPalette で選択された部品を配置したり、それらを接続したりといったことをユーザにさせる。それらの情報は ObjectList に保存される。

アプリケーションエディタを構成する4つの部品は、図1のように接続されている。PartsInfo には部品の名前や使い方、入出力ポートの数と型といった情報が含まれる。PartsID は PartsInfo のメンバのひとつで、それぞれの部品にユニークに付けられた番号である。ObjectList にはオブジェクトの名前や各ポートの接続先、どの部品を使用しているか識別するための PartsID が含まれる。

ユーザは PartsPalette から使用する部品を選択し、Connect に配置する。そして Connect 上の部品を接続するという作業でアプリケーションを作成できる。

3.4 アプリケーションへの組み込み

カスタマイズやアプリケーション開発を容易にするために、アプリケーションからアプリケーションエディタを起動できるようにする。方法は

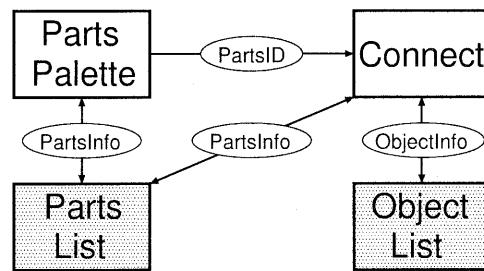


図 1: 構成部品の接続状況

- アプリケーションエディタを呼び出すための部品を使用する、
- アプリケーションの実行時にオプションで指定する、

の二通りがある。

この事でユーザがいつでもそのアプリケーションをアプリケーションエディタで開く事ができ、その編集結果はすぐにアプリケーションに反映されるようになる。

4 まとめ

アプリケーション組み込み型のアプリケーションエディタについて述べている。これにより、GUI アプリケーションのプロトタイピングによる開発が可能になる。

今後、実装を完了し、本プログラム開発方式の開発効率を評価していく。

参考文献

- [1] 大鎌 広, 畠 俊一, 藤原 祥隆: “Partskit: GUI プログラミングのための回路モデル化”, 情報処理学会 全国大会, (1998.3).
- [2] 原 敏弘, 大鎌 広, 藤原 祥隆: “Partskit におけるプログラミングスタイルについての研究”, 情報処理学会 北海道支部連合大会, (2000.11).