

4M-9 制御部品を利用した ユーザインタフェース構築方法

立川 江介 宮下 洋一

(日本電気株式会社)

1. はじめに

最近の計算機技術の進歩によりビットマップディスプレイやマウスを持つ高性能・高機能のワークステーションが急速に普及し、優れた使いやすいユーザインタフェースを求める要求がますます強まってきている^[4]。この要求に応えるために直接操作型のユーザインタフェースが出現したが、その作成にかかる手間がそれまでのものに比べて大きいという欠点を持っていた。これに対して最近には種々のウィンドウシステムが開発され、ツールキットとして提供される部品を利用することによりユーザインタフェース作成の負担が軽減された。しかし部品は画面の構成要素に限られており、制御に関してはアプリケーション作成者が直接プログラムを組む必要があった。

通常のアプリケーションにおけるユーザインタフェースの多くは制御方法が限られているため既存の部品の他に制御用の部品を導入することにより部品の組合せとして実現可能である。ここではインタフェース全体をこのように部品を結合し構築することによりその作成に要する手間を大幅に軽減する方式を提案する。

2. 部品によるインタフェースの記述

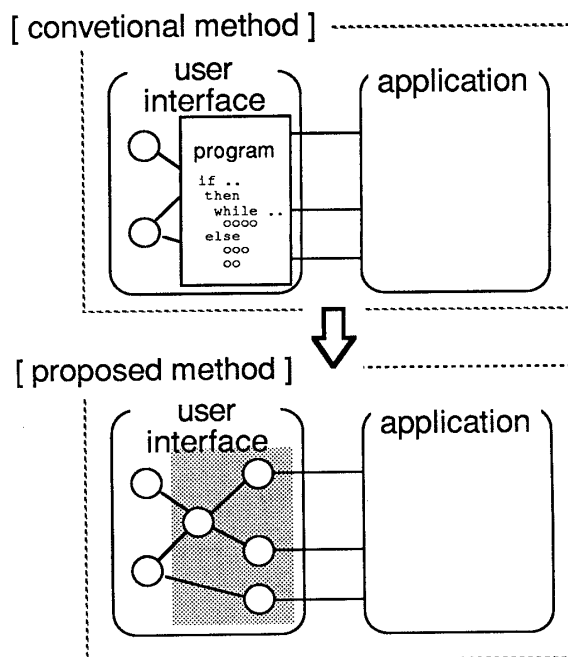
ウィンドウシステムではイベント駆動の考え方が取り入れられ、ユーザの入力はユーザインタフェースで処理された後イベントとしてアプリケーションに送られ特定の処理を起動する形をとることによりアプリケーションとユーザインタフェースの分離が図られている。現在のウィンドウシステムではGMW^[1], Xtoolkit^[2,3]のようにツールキットとして部品を提供し、ユーザインタフェースに関する記述を減らす努力が行われている。しかしこれらのインタフェースでは主に画面の構成要素としての部品のみを扱うにとどまっている。

これに対して、ここではより一般的にユーザインタフェースの制御までも含めた形でユーザインタフェースを部品を結合することにより構成することを考える。これはまた用途をユーザインタフェースの記述に限った一種のビジュアルプログラ

ミングととらえることもできる。

最初にユーザインタフェースをどのような形でモデル化するかについて触れておく。ここではユーザインタフェースをオブジェクト指向の立場からとらえ、インタフェースを構成する部品をオブジェクトと置く。各オブジェクトはメッセージ(イベント)によりお互いに通信する。一つのアプリケーションに限ればユーザインタフェースを構成するオブジェクト、及び各オブジェクトがお互いに通信するイベントの種類は限られたものとなっている。そこでユーザインタフェースを構築する方法としてこれらのオブジェクトをビルディングブロック式に組み立てていく方式が考えられる。

本方式の構成図を図1に示す。現在のウィンドウシステムでは画面の構成要素としての部品しか用意されていないため制御の複雑なアプリケーションに対してはその処理をプログラムとして記述する必要がある。しかし本方式ではユーザインタフェースを制御するための部品を用意することにより制御の流れを部品間の結合として定義する事が可能であり、ユーザインタフェースの構築が容易となっている。



[図1] インタフェース構成

3. 制御を扱う部品の例

ここで制御を扱う部品の例を2つ示す。

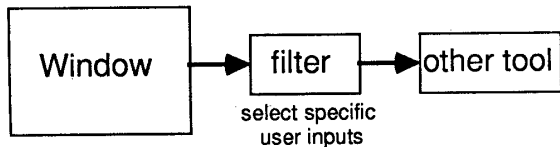
① ウィンドウ内の特定の入力にのみ対する処理

あるウィンドウ内でユーザが特定の入力を行ったときだけ何らかの処理が実行される例を考える。これは特定の入力だけを通過させるフィルタを部品として用意し、それにウィンドウ内で発生するイベントを通しその出力を別の部品へ入力とする事により簡単に記述できる[図2(a)]。複数の部品を組み合わせる際オブジェクト同士はイベントにより通信するが、これは部品間を結ぶ線として表現することができる。

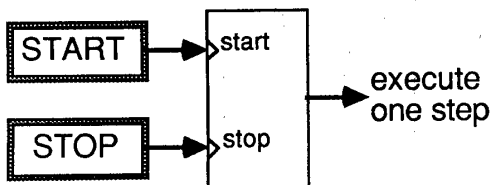
② 繰り返しの制御

繰り返し実行ではユーザが途中でその動作を制御するためにはアプリケーション側での処理とユーザからの入力の間で同期をとる必要が生じる。従来はこのような処理を記述する場合は、アプリケーション側に特別な記述を行う方法か、あるいはユーザインタフェース側の記述言語を用いてプログラミングする方法がとられていた。

ここではアプリケーション側のループの1ステップ分の実行ルーチンをユーザインタフェースから呼び出す形をとる。このオブジェクトはstartのメッセージが到着するとループを開始し、ループ状態ではアプリケーションの1ステップ実行ルーチンをアプリケーションの処理と同期を取りながら呼び出すことを繰り返す。この状態でstopのメッセージが到着するとこのオブジェクトはステップ実行を中止する。図2(b)ではユーザは2つのソフトウェアボタンにより繰り返し実行の開始、終了を指示する。ユーザがSTARTのボタンを選択するとボタンの動作としてイベントが出力され、これにより前述の繰り返し実行制御部品が起動される。同様にSTOPのボタンが選択されると



(a) 特定の入力に対する処理



(b) 繰り返し

[図2] ユーザインタフェース記述例

繰り返しの実行が終了する。

今までのウィンドウシステムでサポートされている部品が画面上に表示される実体のある部品だったのに対し、これらの部品は他の部品が発生したイベントによりシステムの動作を制御する目的を持つ点で異なっている。さらにいくつかの部品を組み合わせる新しい部品を定義することにより部品は階層構成を取る[図3]。

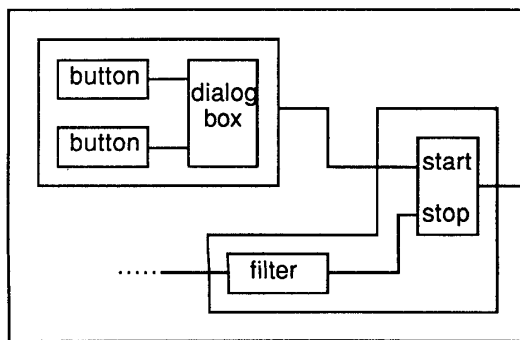
4. おわりに

基本的な部品を組み合わせることによりユーザインタフェースを記述することにより、ユーザインタフェースの作成の手間を大幅に減らすことが期待できる。このためには現在のウィンドウシステムなどで提供されている部品の他にユーザインタフェースの制御を行うための部品を用意する必要がある。ここではこのような制御用の部品としてイベントフィルタと繰り返しを制御する部品について簡単な例を挙げて説明を行った。

今後は種々のアプリケーションに対応するように部品の充実を図るとともに、部品の組合せの手間を簡単にできるようなインタフェースエディタを作成する予定である。

【参考文献】

[1] 大谷 他：ウィンドウ・システム上のアプリケーション構築について、日本ソフトウェア科学会第5回全国大会、B3-5
 [2] McCormack, J. et al., "X Toolkit Intrinsic s - C Language X Interface"
 [3] Swick, R.R., "X Toolkit Widgets - C Language X Interface"
 [4] 本宮, 浜川, 川越：エンドユーザのための視覚的プログラミングシステム-VISOR-, 情報処理学会第36回全国大会、3L-10



[図3] 部品の階層構成