

5 Q-4 鼎(かなえ)インタフェースビルダ「ゆず」のツールボックス機能

太田原剛、†杉山高弘

NEC マイコンテクノロジー(株) ソフトウェア生産技術部、†NEC(株) ソフトウェア生産技術開発研究所

1 はじめに

Xウィンドウ上で動作するアプリケーションプログラムのグラフィカルユーザインタフェース(GUI)開発を効率的に行うツールとして鼎(かなえ)[1]インタフェースビルダ「ゆず」[2]がある。この「ゆず」の特長の一つにツールボックス機能がある。ツールボックス機能とは、ボタン、あるいはメニューなどの対話部品が選択された時に起動される処理(コールバック)を設定するためのものである。

本稿では、以下の特長を持つツールボックス機能について報告する。

- コールバックはマウス操作だけで設定可能
- 設定したコールバックをその場で確認可能
- 設定したコールバックのC言語ソースが自動生成可能

2 ツールボックス機能

ツールボックス機能は以下に示す4種類の処理をコールバックとして設定できる。

1. 標準的処理

鼎のエディタウィジットに対するセーブ、ロード、カット、ペースト、コピー、デリート、ウィンドウのポップダウン、およびアプリケーションの終了の機能を提供している。これらの機能を提供したのは、ユーザがアプリケーションを開発する場合に使用頻度が高いと考えられるからである。

標準的処理の概念を図1に示す。この図に示すように、ユーザはコールバックとコールバック対象を指定するだけで良い。コールバックの具体的な処理はツールボックス機能があらかじめ用意している関数によって実行される。このため、ユーザはコールバックを抽象的にとらえるだけでよい。

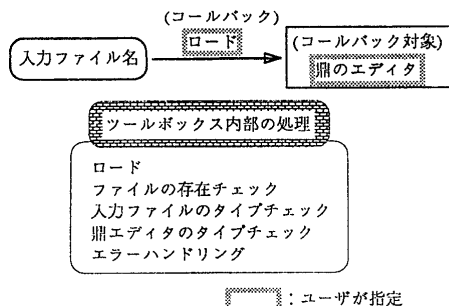


図1. 標準的処理の概念図

Toolbox in Canae Interface Builder "YUZU". Takeshi OOTAHARA, Takahiro SUGIYAMA, NEC Microcomputer Technology, NEC Corporation.

2. 対話部品間操作

フィールドからリストへ文字列をコピー、ボリュームの値をフィールドに取り出すなどのように、GUIを構築する上で考えられる代表的な鼎の対話部品間処理を提供している。これは、3.1節で詳しく述べる。

3. シェルコマンド実行

コールバックでシェルスクリプトを実行する機能を提供している。

4. ユーザ定義関数スケルトン生成

ユーザが既に持つライブラリを使用する場合、あるいはユーザがコールバックのOWNコーディングを行う場合の機能である。この機能は、「ゆず」を用いて新規に作成するGUIとユーザが既に持っているライブラリとのインタフェースを提供するので、ユーザの持つソフトウェア資産を有効に活用することができる。

3 ツールボックスの実現方式

ツールボックス機能では、各コールバックの処理内容を決定するために内部情報を持っている。これを以下の表に示す。

表1. ツールボックスの内部情報

コールバックの種類	内部情報	メンバ名
標準的処理	コールバックの種類 処理機能 対象となる部品	state operation widget
対話部品間操作	コールバックの種類 入力元の部品 出力先の部品	state in_widget out_widget
シェルコマンド実行	コールバックの種類 シェルコマンド 実行結果の出力先	state command exec_out
ユーザ定義関数 スケルトン生成	コールバックの種類 関数名 関数に渡す引き数名 関数に渡す引き数の型	state func_name arg_name arg_type

従って、ツールボックス機能でコールバックを設定するという事は、ユーザからコールバックに必要な内部情報を入力してもらうことになる。

3.1 コールバックの設定

この章では、ツールボックス機能を用いたコールバックの設定方法を、以下の動作をするGUIの作成を例にとり説明する。

1. 所属ボタンをクリックする。
2. 所属一覧リストがポップアップする。

3. 所属一覧リストの中からふさわしい項目を選択する。
4. 設定ボタンをクリックする。
5. フィールドに選択した項目が入力される。
6. 終了ボタンをクリックすると、所属一覧リストがポップダウンする。

この GUI を図 2 に示す。

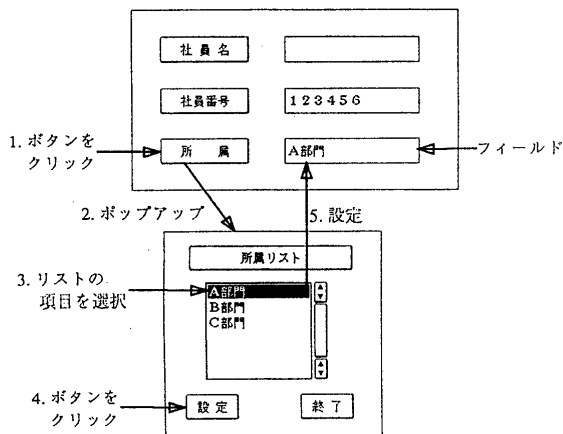


図 2. GUI のウィンドウ例

上記の GUI に必要な処理は、ツールボックス機能の対話部品間操作を用いて設定することができる。例として、設定ボタンをクリックした時のコールバックを設定する画面を図 3 に示す。

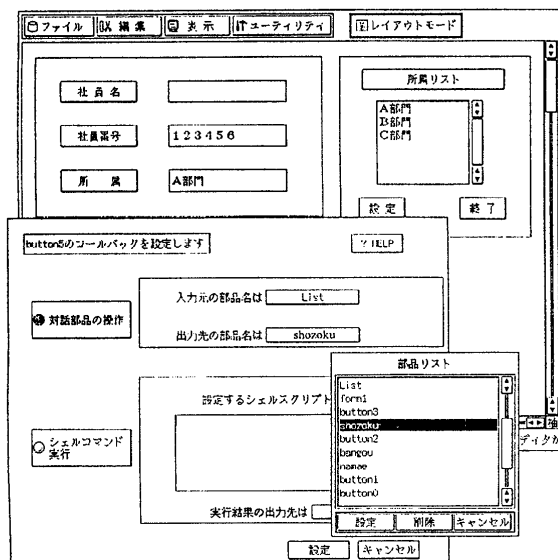


図 3. コールバックの設定

図 3 のウィンドウでは、次の操作を行っている。

- 項目を取り出すリスト (入力元の部品) を指定する。
- 項目を表示するフィールド (出力先の部品) を指定する。
- 指定するリストおよびフィールドは、部品リストから選択する。

このように、ユーザは入力元の部品名と出力先の部品名だけをマウス操作で指定することによって、ノンプログラミングでコールバックを設定でき、入出力部品間のデータ変換はツールボックス内部で自動的に決定している。

3.2 コールバックのリハーサル

ツールボックス機能を用いて設定されたコールバックは、「ゆず」の中で確認 (リハーサル) を行うことができる。リハーサルは、コールバックが設定された対話部品が持つ内部情報を解析して実行される。このため、ユーザはコールバックを抽象的にとらえるだけでよい。

これにより、ユーザは「ゆず」内部でコンパイルすることなく作成した GUI の振る舞いをその場で確認することができる。

3.3 コールバック記述の自動生成

ツールボックス機能を用いてコールバックが設定されたボタンあるいはメニューなど対話部品に対して、コールバックを設定するための C 言語の記述を行う。

コールバックの記述に関しては、コールバックの処理内容をツールボックスで提供している場合とユーザ定義関数スケルトン生成とで異なり、それぞれ以下ようになる。

3.3.1 機能を提供している場合

標準の処理、対話部品間操作、およびシェルコマンド実行のコールバックはツールボックス機能で処理内容を提供している。従って、これらの処理に必要な C 言語ソースは、ボタン、メニューなどに対してコールバックを設定する記述および処理内容の記述である。このうち、ツールボックス機能で提供している処理の内容は固定であるため、処理内容を記述したファイルを提供している。このため、C 言語ソース生成時には、コールバックが設定されたボタン、メニューなどにコールバックを設定するための記述を生成すればよい。

3.3.2 ユーザ定義関数スケルトン生成

ユーザ定義関数スケルトン生成では、対話部品に対するコールバックの設定と、コールバック関数のスケルトンを生成するようにする。スケルトンの内部は、ユーザに記述してもらうようにする。

4 まとめ

「ゆず」のツールボックス機能を用いることにより、アプリケーションソフトの開発者が簡単にコールバックを設定できることを示し、ツールボックス機能の実現方式を述べ、ツールボックス機能の有効性を確認できた。

参考文献

- [1] 暦本, 他: "X ウィンドウ上のマルチメディアユーザインタフェース構築環境: 鼎" 情報処理学会第 30 回プログラミングシンポジウム予稿集, 1989.
- [2] 杉山, 他: "鼎 (かなえ) インタフェースビルダ「ゆず」の構築" 本予稿集.