

グラフィカルユーザインタフェース構築ツール「ゆず」

杉山高弘* 伊藤博司* 宮内幸司* 太田原剛** 惣門真彦**
 新井英俊*** 児玉 康行***

*NEC ソフトウェア生産技術開発研究所,

NEC マイコンテクノロジー(株), * 神戸 NEC ソフトウェア(株)

概要 アプリケーションシステムの GUI を開発する際に、グラフィカルユーザインタフェース (GUI) 構築環境「鼎 (かなえ)」の部品を対話的にレイアウトし、C 言語の GUI プログラムを自動生成するグラフィカルユーザインタフェース構築ツール「ゆず」を紹介する。「ゆず」は製品化を完了し、多くのユーザに利用されている。

「ゆず」はマウスによって対話的に、GUI 部品 (ボタン、メニュー等) のレイアウト、テキスト、色等の属性やサブメニュー・ポップアップ関係を設定し、その場で確認できる。特に、ボタン等が選択されたとき実行される処理記述部に対しても、あらかじめ内部定義してある処理関数を GUI 画面から設定する機能 (ツールボックス) を提供している。これらの機能の実現方法と有効性を示す。

「ゆず」を開発に用いれば、1) GUI 仕様の設計段階で最終イメージの早期明確化、2) プロトタイプシステム開発の効率化、3) 生成された GUI プログラムの品質向上、が実証できたことを報告する。

Graphical User Interface Builder: YUZU

Takahiro Sugiyama*, Hiroshi Ito*, Kouji Miyauchi*, Takeshi Ootahara**, Masahiko Soumon**

Hidetoshi Arai***, Yasuyuki Kodama***

*Software Productivity Engineering Laboratories

**NEC Microcomputer Technology, Ltd.

***NEC Software KOBE Ltd.

This paper discusses the Graphical User Interface (GUI) Builder "YUZU". When a user develops GUI, the YUZU assists to design GUI and generates GUI C source programs using CANAE: a platform for constructing graphical user interfaces with editors. The YUZU has already put on the market and is available for many users.

The YUZU interactively lays out GUI parts (button, menu, etc.) and sets their attributes of text, color, submenu, etc. The user can try out and evaluate the designed GUI at the same time. The YUZU provides the TOOLBOX, which can set callback procedures of menu, button, palette, etc. without programming, by preparing pre-defined procedures.

A GUI development using YUZU has the following merits: 1) to realize a final image of a GUI design in a design phase, 2) to improve efficiency developing a prototype system, 3) to improve the GUI program quality.

1 はじめに

アプリケーションシステムのインタフェースに、近年、マルチウィンドウやアイコンを駆使した操作性と見栄えが優れたグラフィカルユーザインタフェース（GUI）が求められてきている。そのような GUI を開発するには複雑なウィンドウシステムの理解と膨大な開発期間を必要とする。特に GUI の分野では、最終イメージを実現するために机上設計、製造、評価を繰り返さなければならなかった。

グラフィカルユーザインタフェース構築ツール「ゆず」[1]は、グラフィカルユーザインタフェース構築環境「鼎（かなえ）」[2][3]の GUI 部品を利用して、アプリケーションシステムの GUI を対話的に設計支援し、その結果から完全な C 言語の GUI プログラムを自動生成する。

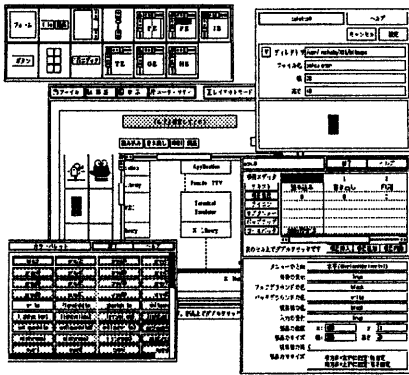


図 1: GUI 構築ツール「ゆず」の画面

「ゆず」を利用すれば、CASE, CAD, CAE やデータベース検索結果の表示部等を始めとする様々なソフトウェアの操作性を格段に向上させる GUI を開発できる。GUI の設計段階で最終イメージを確認できるため改工数が減少できる。また、「ゆず」は文字入力を除いてマウスのみによって操作可能のため GUI 開発期間を大幅に短縮できる。

2 ゆずの概要

「ゆず」(図 1)は、図 2 で示すように UNIX¹上のウィンドウシステムとして最も普及している X ウィンドウシステム²上で構築された GUI 構築環境「鼎（かなえ）」³によって構築されている。鼎は、ボタン、メニュー、パレット、フィールド、スクロールバー、リストの対話部品とフローチャート図や図形等を描画できる 6 つのエディタ部品から構成されている。

開発中の GUI に対する設定や変更がリアルタイムに視覚的に認識できるほうがより GUI 開発をスムーズにするといわれている。これを実現するために、Common Lisp のサブセットの処理系に約 1300 の鼎および X ウィンドウの関数を登録した鼎 Lisp[4]を独自に開発した。「ゆず」は、この鼎 Lisp によって記述されている。

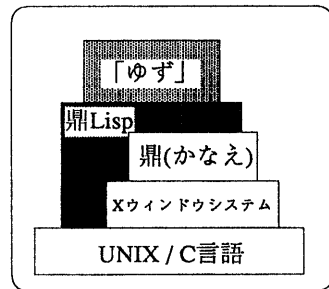


図 2: 「ゆず」の構築環境

図 3 に「ゆず」のシステム構成を示す。以下に各構成部の説明をする。

● 部品パレット

鼎の対話部品（ボタン、メニュー、パレット、フィールド、スクロールバー、リスト）とエディタ部品（テキスト、表、図形、グラフ構造、階層構造、イメージ）とフォーム部品の

¹UNIX は、UNIX System Laboratories, Inc. が開発しライセンスしています。

²X ウィンドウシステムは、MIT の商標です。

³鼎（かなえ）は、NEC の商標です。

合計 13 種類の部品から選んで組み立てることによってアプリケーションの GUI を開発できる。

- 部品配置部

鼎のグラフ構造エディタで実現されており、レイアウト管理機能を持っている。部品配置部は、台紙のような画面に、部品パレットから選んだ部品を張り付けていくことによって開発したい GUI をレイアウトする。部品のレイアウト、リサイズ、部品間の親子関係、画面のスクロール、再表示等を行なえる。

- リソースエディタ

配置した部品の属性（表示タイトル、表示項目数、色、ピットマップ等）を入力文字列以外すべてマウスによって変更できる。設定項目はすべて GUI 画面で提示する。次の操作に戸惑いそうなところや間違った設定では、システム側が進んで操作の手引を行うので、初心者やプログラマ以外の人でも GUI を開発することができる。このリソースエディタが「ゆず」をより一層使いやすくしている機能である。

特に、本リソースエディタを使えば、鼎 C ライブラリや X ツールキットの深い知識がなくても部品の属性のカスタマイズができる。

- リハーサル機能

GUI 構築ツールの一つの重要な機能である設計した画面をその場で確認する機能である。サブメニュー、ウィンドウのポップアップ、ツールボックスによって設定した処理をその場で評価することができる。

- ツールボックス

「ゆず」の配置部品間の処理のうち汎用的なものに関して、ツールボックスがあらかじめ用意した関数を使用できる。利用者は

必要最低限な情報を入力するだけで、リハーサル機能によってその場で試すこともできるし、設定した処理を再現する C プログラムも生成する。

- ソース自動生成 / 実行機能

設計した画面から鼎ライブラリを用いた C 言語の GUI プログラムを自動生成する。生成されたプログラムにはコンパイルエラーがないし、優秀な GUI プログラマが開発したプログラムと較べても効率上特に問題はない。自動生成した GUI の C プログラムをライブラリ等の環境を気にせず気軽に「ゆず」の中からコンパイル、実行できる。完成物件を確認するのに役立つ。「ゆず」の環境から巣立ったコンパイルオブジェクトは、ポータビリティに優れる。

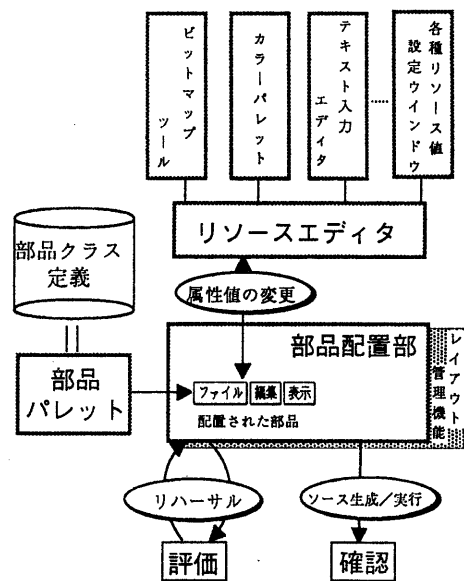


図 3: 「ゆず」のシステム構成

「ゆず」の特徴の中から以下の 2 点に絞って、本稿では説明する。3章では、GUI 構築ツールの最も基本となるレイアウト管理機能に関する構築

方法について説明する。4章では、ツールボックスと呼ばれるノンプログラミングでのコールバック処理の設定機能について説明する。

3 レイアウト管理機能

「ゆず」の中で最も基本となる GUI 部品を配置するための機能 (レイアウト管理機能) について、実現の方式とその利点を説明する。

3.1 鼎エディタを用いたレイアウト管理機能

レイアウト管理機能としては、主に以下の項目が挙げられる。

- 設計画面を構成する部品の自由な移動、拡大・縮小
- 設計画面を構成する部品の親子関係の設定、変更
- 設計画面のスクロール、再表示、整列機能
- ユーザのマウス入力を処理し、上記機能の呼出

これらを実現するためには、複雑なマウスイベントやウィンドウの制御を行わなければならない。「ゆず」では鼎のグラフ構造エディタを利用して、部品をレイアウトし編集する部品配置部を実現した。これによって、上記処理の複雑な処理部分の大部分をグラフ構造エディタが有する本来の機能で代行できるため、上記処理機能の実装を「ゆず」自身の中から排斥できる。

「ゆず」で実際に鼎部品が配置されるとき、部品クラス定義に従って部品インスタンス (以後、部品と呼ぶ) を作成する。部品は図 4 で示すように、鼎エディタ層の見えないノードと可視層 (ビュー) の鼎エディタの子として部品の実体を作成される。鼎の見えないノードの直上に部品の実体が重なっているイメージを表す。

最初に述べたレイアウト機能を鼎エディタの機能で代行させるためには、マウス操作は部品の実

体を透過して鼎のノードへ達する必要がある。ただし、実際に設計中の GUI の動作を試すリハーサル機能では、マウスが部品の実体を操作できないなければならない。

このため、「ゆず」では部品の実体がマウス操作を透過させたり、つかまえたり自在に切替える仕組みを実現した。

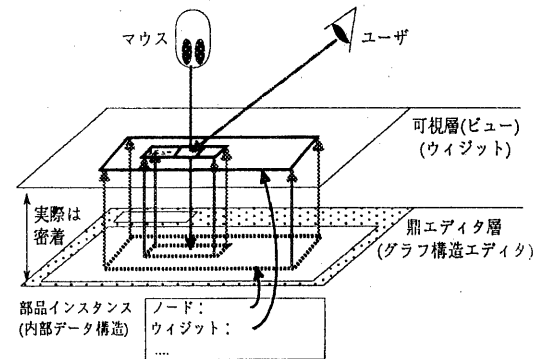


図 4: 部品配置ウィンドウの実装

さらに、鼎エディタ層のノードと対応する可視層のウィジェットをマウスによるリサイズや移動、またはリソースエディタからの属性変更にとまらうリサイズに関して連動させるための制御を組込む。このとき、鼎エディタ層が提供している鼎イベントマップ機能を利用する。鼎イベントマップは、発生したマウスイベントの種類によって予め処理を登録し、そのイベントが発生した際に対応する処理を実行する。

図 5 にマウスによって部品を拡大する時の処理の流れを具体的に図解した例を示す。まず、鼎エディタ層の図形要素がマウスによって拡大される。これは、鼎イベントマップの図形要素のリサイズ処理が行なうが、この時同時に、可視層のウィジェットを図形要素にあわせてリサイズする連動処理記述が呼び出される。

本方式を採用した利点の 1 つは、他の GUI スタイル (Motif⁴, OPEN LOOK⁵等) を用いたインタフェースビルダへの改造の際も可視層のウィジッ

⁴Motif は、OSF の登録商標です。

⁵OPEN LOOK は、AT&T の登録商標です。

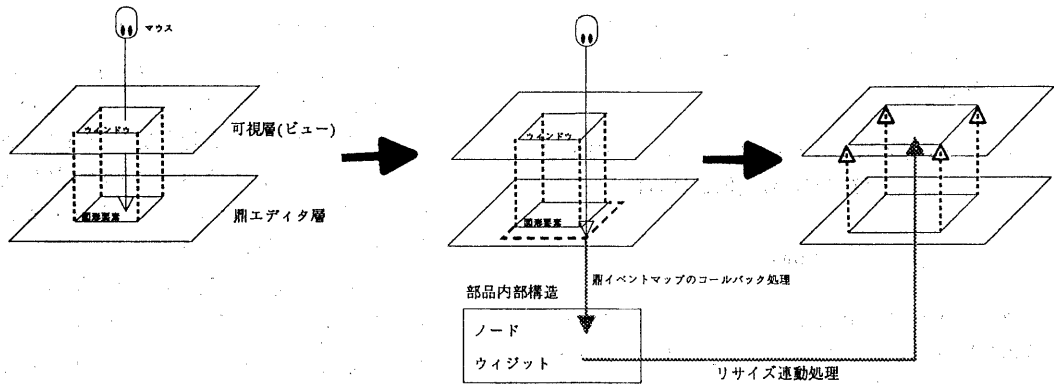


図 5: リサイズ時の連動制御例

ト作成部のみを改造すればよく、鼎エディタ層の機能は変更せずそのまま流用できることである。

3.2 リハーサル機能

インタフェースビルダのもう一つの重要な要件として、設計した画面をその場で確認するリハーサル機能がある。例えば、サブメニューやポップアップウィンドウの一部のウィンドウは、初期画面には表示されず、特定の条件下のみ表示される。リハーサル機能はこれらの動作を確認するのに用いられる。ただし、GUI構築ツールにおいて、「レイアウト時には、特定条件下のみで表示されるウィンドウも直接操作できる状態で配置し、属性の変更を確認したい」という要求がある。

「ゆず」では、図 6 で示すようにレイアウト時には、サブメニュー等の親部品を鼎エディタ層のグラフ構造エディタにすることによって疑似状態を作る。これによって、特定の条件下のみ表示されるはずの部品を常時表示できる。リハーサル時には、部品の実体を通常の状態に戻すとともに、サブメニューやポップアップウィンドウ等の本来の親部品へ変更することによって、実際の動作を実現できる。

本方式の利点として以下が挙げられる。

1. 設計画面を構成する部品の自由な移動、拡

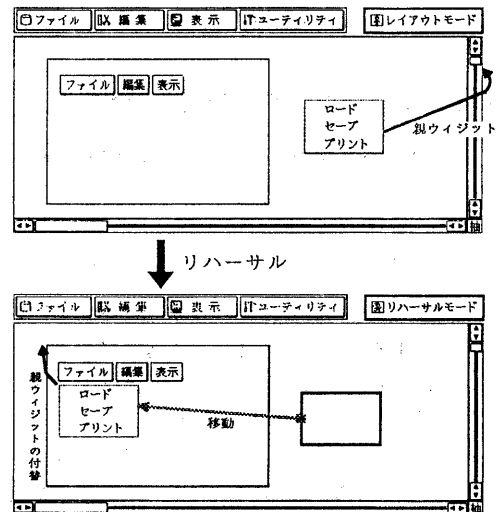


図 6: リハーサル機能

大・縮小、部品間の親子関係の設定、変更、スクロール、再表示、整列機能等を鼎エディタの既存機能に依存可能

2. X ツールキット等のウィジェットの作成順序等の制約を受けずに、サブメニューのウィジェットから作成したり、ユーザの思いのままの設計手順を踏むことが可能
3. リハーサルへ変更する際、ウィジェットを作

り直さずにすむので、スピード、メモリ使用量の面から効率良くリハーサル状態へ移行可能

4. Motif 等の異なる GUI スタイルの GUI 構築ツールへの移植が比較的簡単に可能

4 ツールボックス機能

ツールボックス機能 [5] とは、ボタン、メニュー等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) を対話的にノンプログラミングで設定できる。「ゆず」のリハーサルモードで設定した処理をその場で確認もできる。設定したコールバックは、ツールボックス利用者に C 言語ソースプログラムとして提供される。

4.1 ツールボックス機能の分類

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

GUI 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標準的処理を設定する。現在、鼎のエディタウィジェ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。

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、ボリュームの数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。この実例と設定画面を 4.3 節で説明する。

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する。

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関数のオウンコーディングを行う場合の機能である。ツールボックスのユーザ定義関数設定ウィンドウから、コールバック関数名と引数、および、引数の型を入力すれば、コールバック関数の設定とコールバック関数のスケルトンをソースプログラムへ生成する。

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。

4.2 ツールボックスの設定モデル

ユーザは処理される対象とその処理の種類 (コールバック) を指定する。コールバックの具体的な処理記述はツールボックス機能があらかじめ「ゆず」内部に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処理内容を抽象的に捉えさえすればよい。

例えば、図 7 に示すように、ボタン部品のコールバックとして鼎のエディタ (処理対象) ヘファイルのロード (コールバック) 処理を設定しようとしたとき、ユーザはツールボックスを用いれば、鼎のエディタとロードをマウスで選択するだけである。

これをリハーサルする場合、「ゆず」にあらかじめ登録しておいたロード関数がファイルの存在、エディタとのタイプチェック、エラーメッセージ出力、ファイルのロード等をすべて実行する。これらの処理の一つ一つをユーザは知らなくてもすむ。

ソース生成時は、ツールボックスで設定したコールバックの内部関数に相当する関数を外部ファイル (C ソースプログラム) として「ゆず」が提供している。リハーサル時とソース生成時で動作が全く同じであれば、同一関数を呼び出す。GUI 開発途上に一時的に発生する不正な設定をチェック

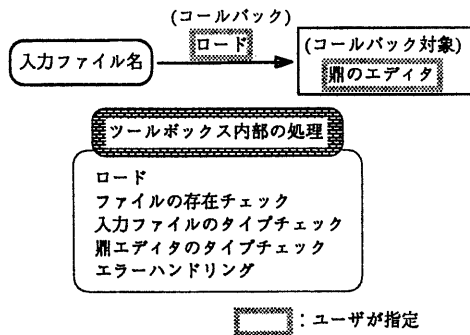


図 7: 標準的処理の概念図

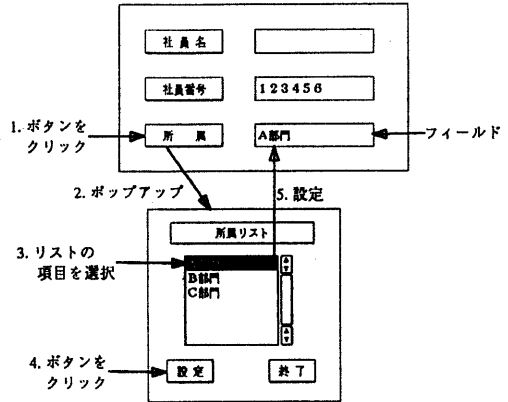


図 8: ツールボックスの応用例

する機能拡張を施した内部関数としてソース生成時の関数とは独立に登録もできる。

4.3 ツールボックスの実用例

ツールボックスを利用すれば、図 8 に示す画面と以下に示す処理すべてをマウス操作のみで実現できる。ツールボックスを用いて処理を設定する入力画面例を図 9 に示す。

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。

上記の動作は、各々のツールボックス機能を用いて設定することができる。例として、設定ボタンを選択した時のコールバックを設定する対話部品間操作コールバックの設定画面を図 9 に示す。

図 9 の設定画面では、次の操作を行う。

- 項目を取り出すリスト (入力元の部品) を指定
- 項目を表示するフィールド (出力先の部品) を指定
- 指定するリストおよびフィールドを部品リストウインドウから選択 (部品リストウインドウは、現在配置されている部品名の一覧を表示する)

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する。

5 おわりに

「ゆず」は、EWS4800 (CISC, RISC), SUN4 の X11R4 および X11R5 で稼働する。プロトタイプ版「ゆず」を用いて最終版「ゆず」を開発した結果、全体規模の約 25 % を自動生成することができた。この部分に関してはほとんど工数を必要としなかったため、開発期間を 1/4 だけ縮められた。他の CASE ツール等を調査したところ、大規模な CASE ツールにおいても、やはり全体の 1/4 ~ 1/5 くらいが「ゆず」によって自動生成できる

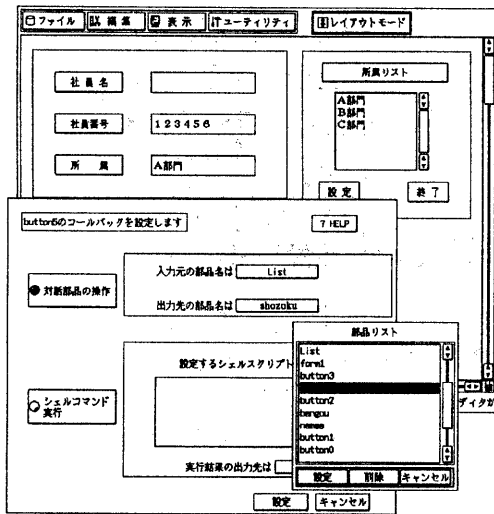


図 9: 対話部品間操作コールバックの設定画面

と思われるので、同様な効果を期待できる。GUIを重視した分野のアプリケーションは、GUI記述部が1/2以上にもものぼるといわれ、「ゆず」の効果がさらに明確になる。

今後は、他人数でGUIを分散開発する場合の支援としてGUI部品登録・再利用機能、GUI部分以外の処理に関するOWNコーディング記述の保存、Motifインタフェースビルダの研究開発を行なっていく予定である。

最後に、終始暖かい助言を頂いたソフトウェア生産技術開発研究所のみなさまに感謝します。プロトタイプ版「ゆず」の研究開発に多大なる貢献をされたNEC System Laboratoriesの立川氏へ深謝します。

参考文献

- [1] 杉山他: "鼎(かなえ)インタフェースビルダ「ゆず」の構築", 情報処理学会第45回全国大会 5Q-3, 1992
- [2] 暦本, 他: "エディタを部品としたユーザインタフェース構築基盤: 鼎" 情報処理, 第31巻第5号, 1990

- [3] 秋口, 他: "ユーザインタフェース構築環境: 鼎(かなえ)" NEC技報 Vol.45 No.3, 1992
- [4] 杉山, 他: "ユーザインタフェース構築環境「鼎」のスク립トインタプリタ" 情報処理学会第39回全国大会, 1989.
- [5] 太田原, 杉山: "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", 情報処理学会第45回全国大会 5Q-4, 1992