ヒューマンインタフェース 47－7
（1993. 3. 1）

AutoView:プログラミング知識を必要としないUI構築ツール

吉山 正治，佐藤 武秀，増田 英孝，笠原 宏
(東京電機大学　工学部)

概要

筆者らは，対話的なグラフィカルユーザインタフェース(GUI)の構築を容易に実現するための統合的な支援環境の研究を行なっている。その一つとして，GUIを利用したことがあり，GUIを構成する部品の意味や動作についての知識を持つが，GUIのプログラミングの知識を持たないユーザにも利用できるようなGUI構築ツールの対話方式について検討を行なっている。

本稿では，サンプルの提示，属性の設定，ヘルプシステムを利用したユーザとの対話方式を提案し，これらの概念に基づくUI構築ツール**AutoView**について紹介する。さらに**AutoView**はインタフェースの作成だけではなく，インタフェースの振る舞いを記述することが可能である。

AutoView:An UI builder without the knowledge of GUI programming details

Masaharu YOSHIYAMA,Takehide SATO,Hidetaka MASUDA,Hiroshi KASAHARA
Department of Electrical Engineering,
Faculty of Engineering,
Tokyo Denki University,
2-2 Kandanishiki-cho, Chiyoda-ku,
Tokyo 101, Japan

Abstract

We have been involved in the research of an integrated environment for simplifying interactive GUI (Graphical User Interface) construction. One of our active research areas is an interactive UI builder tool for those end users who are familiar with the manipulation and meaning of GUI components but ignore GUI programming details.

We propose an interactive dialog method for users to select samples, set attributes, and use helps. **AutoView** can also specify not only user interface but also interface behavior by transferring interaction objects's attributes each other.

1 はじめに

筆者らは，対話的でグラフィカルなユーザインタフェース (GUI) の構築を容易に実現するための統合的な支援環境の研究を行なっている。その一つとして，GUI を利用したことがあり，GUI を構成する部品の意味や動作についての知識を持つが，GUI のプログラミングの知識を持たないユーザにも利用できるような GUI 構築ツールの対話方式について検討を行なっている。

視覚的，直接的にアプリケーションの操作を行なうことができる GUI は，ユーザに操作性の向上，効率の向上をもたらす。しかし，GUI を持つアプリケーションの開発は，非常に複雑である。その理由として，ユーザとの対話定義を行なう部分の作成負担が大きい [1]，またその対話定義を記述するためのツールキットの使用方法が複雑である等が挙げられる。

この問題を解決するために，さまざまな GUI 構築ツールが開発されている。これらのツールは，マウスなどのポインティングデバイスで，作成する部品を選び出すことにより視覚的にレイアウトを作成することができる。これを使用することにより，アプリケーションの作成の効率化をはかることができる。

しかしこれらのツールを使用するためには，部品の詳細や，階層，リソースなどのウインドウプログラミングに関する専門的な知識が必要である。

そのため，GUI プログラミングの知識のないユーザがこれらのツールを用いて，自分の目的とする GUI を作成することは非常に難しい。

また，これらの GUI 構築ツールは，対話定義の作成のみがほとんどであり，対話制御やアプリケーション本体については，プログラミングを行なわなければ作成できない。

本稿では，このような GUI の専門的知識やプログラミングの知識を持たないエンドユーザが GUI の構築，対話制御の作成を行なうための方針および対話形式について述べ，それに基づき開発を行なっている **AutoView** の概要について述べる。

2 GUI 構築ツールの問題点

GUI 構築ツールは，GUI プログラマのアプリケーション作成負担を軽減する目的に作られているものが多く，少なくともツールキットを用いたアプリケーションの基本的な作成手順や，部品の基本的な知識が必要である。

そのため，実際に GUI プログラマでないユーザが，これらの GUI 構築ツールを用いてアプリケーション開発するには，基本的知識を含めて以下のような知識が必要となる。

- 部品の種類
  ユーザはそのツールに用意されている部品の Look&Feel，またその動作についての詳細を知らなければならない。
- 部品の階層
  GUI を作成する場合に部品をどのように組み合わせてゆくのか，またどの部品とどの部品とを組み合わせてゆかなければならないのか。
- リソース
  部品の色や，フォント，配置などの，アプリケーションをカスタマイズするリソースの設定を行なうための知識。

GUI 構築ツールの多くは，ツールキットをベースに作成されているので，ツールキットと同様な知識を必要とする。

多くの GUI 構築ツールはレイアウト部の作成のみであり，対話制御部の作成は行なわれない。そのため，UI 構築ツールで作成されたソースコードに対して，プログラミングをしなければならない。最終的に 1 つのアプリケーションを完成させるにはさらに以下のような知識が必要である。

- 言語知識
  プログラミングにはGUI構築ツールに用意されている専用ライブラリ，もしくはツールキットを用いてプログラミングをしなければならない。ツールによっては専用の言語を用いなければならない。

3 AutoViewの設計指針

3.1 ユーザインタフェース作成支援

前節で述べたように，GUIの作成をUI構築ツールを用いて作成することは専門家以外では困難である。ではエンドユーザの持つ知識でGUIを作成するにはどのような手法を用いればよいかを検討する。GUIを利用しているユーザの知識は以下のものがあげられる。

1. GUIの操作方法が理解できる。
2. GUIに用いられている部品の意味がわかる。
3. プログラミングの知識がない，もしくはGUIプログラミングの知識がない。
4. アプリケーションの使用方法がわかる。

このユーザの知識でGUIの作成を可能するために，**AutoView**では以下のような機能を提供する。

部品の階層の変更

ツールキットを用いてGUIのレイアウトの作成を行なうには，各部品間の階層を考えて作成しなければならない。その階層構造を意識せずにユーザにレイアウトの作成が行なえるような機能を提供する。

これらの実現には部品の階層を変更するのではなく，ユーザに提供する階層と，実際の部品の階層との異差を，アプリケーションが吸収することで実現する。この階層をツールキットレベルで変更してしまうと，逆に，ソースレベルでの変更を行なう場合に，専門的知識を持つGUIプログラマの作業効率の低下をまねいてしまう。

サンプルの提示

ユーザが扱うことのできる部品の一覧をユーザに提示する。これにより，ユーザは，その部品のLook&Feelや，動作，使用目的を確認することができる。

属性の設定を最低限のものにする

それぞれの部品は，さまざまな属性を設定することによって，初めて使用することが可能となる。部品によりその属性の設定は数十種類設定にも及ぶ。そこで，ユーザがGUIを作成する場合には，各部品の必要最低限の属性を設定させ，その他の属性は，システムのデフォルト値を設定する。これにより，複雑な属性の設定や，ほとんど用いられることのない属性の値を気にする必要性がなくなる。

3.2 対話制御部作成支援

実際のGUIでは，ユーザがボタンなどの部品を操作した場合に呼び出すコールバックが数百にも及ぶ場合がある。しかし従来のGUI構築ツールはGUIの定義のみであり，この膨大で，複雑なコールバックをツールキット，もしくは専用のライブラリなどを用いて作成しなければならなかった。そのためコールバックの作成の負担の軽減は，ほとんどはかられていないのが実状である。コールバックの作成の負担の軽減のために，スクリプトなどが導入されているものもあるが，それらの言語を習得するまでに時間がかかるなどの問題点がある。

コールバックは作業の目的によって決まるので，コールバックの記述はGUIの構築とは違い一般性がない。そのためプログラミング

なしでコールバックの作成をすることは困難である。NeXT Interface Builder[5] のように，コールバック関数をあらかじめ用意しておく方法が考えられるが，自分の目的とするコールバック関数がリスト中に見つからなければ自分で作成しなければならない。

AutoView ではコールバック関数をある程度機能を限定し内蔵し[3]、さらに属性からのコールバックの作成を行なう。コールバックを自動的に作成する手法としては，各部品の属性に注目し，部品間の属性の受け渡し，アプリケーション本体へ渡す属性とアプリケーションから返ってきた属性の受け渡しによって行う[2]。

コールバック関数

コールバック関数は使用する部品によってある程度決定される。例えばテキストのウインドウが作成された場合，そのコールバック関数として必要なものはファイルのセーブ，ロード，ウインドウ内の消去などである。このように使用目的の明確な部品に対し，決まったコールバック関数を用意することにより，ある程度作成負担を軽減できる。

属性によるコールバックの作成

上述の3つの部品に対し，ある決まったコールバック関数を用意できない部品，ここでは，ボタン，メッセージ，リスト，チョイス（選択），トグル，スライダ，テキストに対するコールバックの作成方法について述べる。

GUIを構成している部品は，部品ごとに座標や，ラベルなどの様々な属性を持っている。この属性はGUI作成時に用いられるだけではなく，他の部品から参照したり，されたりしている。例えば，表計算のようなアプリケーションでは，各セルの値は自分の持っているセルの値や，他のセルの値を用いて計算される。このように，各部品の持っている属性の受け渡しや，その値の加工を考えることによって，コールバックを自動的に作成することができる。

実際に部品の属性を用いてコールバックを作成するために，各部品の動作を明確にしなければならない。そこで属性の受け渡しに関する動作を以下の3つに分類して考える。

1. **自分自身に対して作用する**
   自分を動作させた時，自分に対して作用するものである。
2. **他の部品から作用を受け，属性の受け渡しをする**
   他の部品が動作させられた時，自分がその作用を受け，属性が変化するものである。
3. **自分以外の他の部品に作用し，属性の受け渡しをする**
   自分自身が動作させられた時，他の部品を作用させるものである。

既存のアプリケーションの利用

ウインドウシステムでは，すでに多くのアプリケーションが用意されている。ユーザがアプリケーションを開発する場合，アプリケーションの全てを開発するのではなく，この既存のアプリケーションを活用することにより，開発作業の軽減をはかることができる。例えば，作成するアプリケーションに，エディタや，プリントアウトのアプリケーションが必要な場合，すでにこれらのアプリケーションがウインドウシステムで用意されていれば，その部分の作成は行わなくとも，そのアプリケーションを呼び出すことで解決する。

4 AutoViewの概要

AutoView をOpenWindows上にXView[6], X-libを用いて構築した。図1に **AutoView** の起動時の画面を示す。**AutoView** は，部品

一覧ウインドウ（左）と作成ウインドウ（右）から構成されている。ユーザがレイアウトを作成するには，部品一覧ウインドウから部品を選び出し，作成ウインドウにその部品を配置してゆくことにより行なわれる。その時，各部品の属性の設定およびコールバックの作成を行なってゆく。

AutoView は，部品を中心にして，レイアウトの作成支援，対話制御の作成支援を行なう。作成された部品に対し，コールバックを結合，またはコールバックを自動生成し，最終的に XView を用いたソースコードを生成する。この時ユーザが作成したソースコードを結合することも可能である。それぞれ作成された部品は，View ファイルとして保存される。画面のレイアウトだけではなく，コールバックの情報を含んでいる。

最終的に生成されたソースコードは，コンパイルすることにより **AutoView** 上ではなく，OpenWindows 上で直接実行することができる。そのため，作成したアプリケーションのパフォーマンスを低下させずに実行することが可能である。

4.1 AutoView の機能

前節に基づき実装した **AutoView** の機能について述べる。

4.1.1 レイアウト作成支援

画面のレイアウトは，部品が格納されている部品一覧ウインドウから，部品を選択し，作成ウインドウに配置することによって作成される。部品は，図1に示めすように，3種類のウインドウ (キャンバス，シェル，テキスト)，および，7種類のアイテム (メッセージ，ボタン，選択，スライダ，トグル，テキスト，リスト) から構成されている。この１０種類の部品間には階層関係がないため，ユーザは部品の階層構造を気にせず，作成ウインドウに対し部品を配置してゆくことが可能である。

また，作成する部品の属性は，属性設定ウインドウ上でマウスとキーボードを用いて設定することができ，属性の再設定時には見た目の変化を画面上ですぐに確認することができる。

図2にボタンの属性設定ウインドウの画面を示す。

ツールキットでは，各部品にごとに細かい属性の設定を行なうことが可能であるが，**AutoView** では使用頻度の低い属性については，その値をシステムのデフォルト状態に設定する。これにより，ユーザの属性の設定数を減らすことができ，作成負担を軽減することができる。

各部品についての詳細がわからない場合は，部品一覧ウインドウ上の" サンプル表示" ボタンを押すことにより，その部品の Look&Feel および動作を確認することができる (図 3)。このサンプル一覧ウインドウの部品は実際に動かしたり，編集等を行なうことができる。アプリケーションに適用された場合と同様の Look&Feel，動作を確認することにより，ユーザは，レイアウト作成時に各部品の仕様が明確になる。これにより，ユーザは部品を容易に取り扱うことができる。

スクロールバーなどの単独で使われることのない部品は，スクロールバーが付随可能な部品を選択した時に，作成するかどうかをユーザに問い合わせるようになっている。

4.1.2 対話制御部作成支援

コールバック関数

テキストウインドウ，キャンバスウインドウ，シェルウインドウなどの部品は他の部品に作用することはないので，各ウインドウの処理に応じたコールバックを用意する。

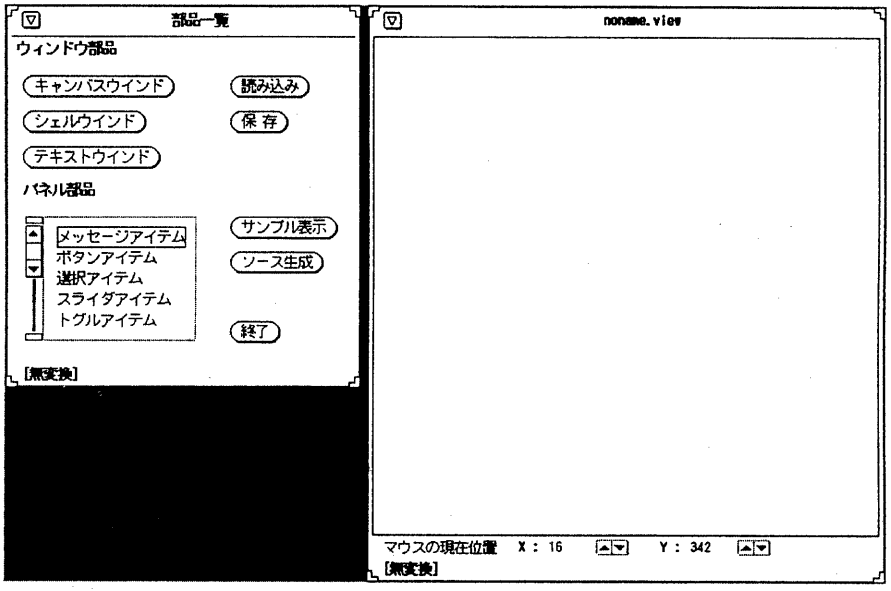

図 1: AutoView 起動時画面

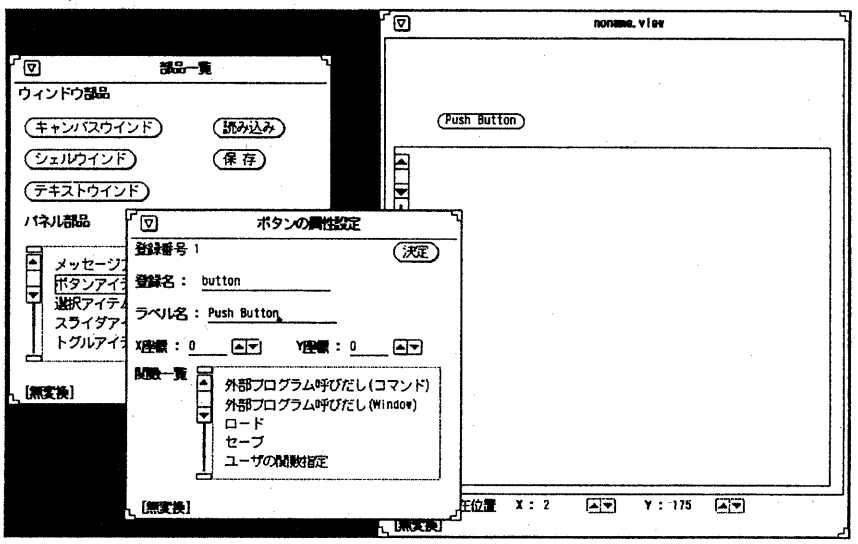

図 2: ボタンの属性設定

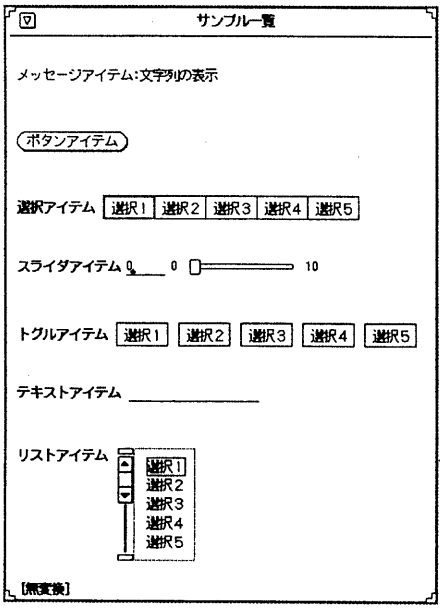

図 3: 部品のサンプル

属性によるコールバック作成

属性の受け渡しによるコールバックの指定は以下の手順で行なわれる。

1. 部品を作成する。属性の受渡しを行なう部品が作成されていない場合は，相手の部品を作成する。
2. 作成済み部品一覧ウインドウから，対象となる部品の登録名を指定する。
3. 部品から受け取る，もしくは受け渡す属性を指定する。

属性を受け渡す際に，その属性の加工が必要であれば，アプリケーションに渡す属性と，アプリケーションからの戻り値の部品の指定を行う。

4.2 エラーハンドリング

作成されたアプリケーションが実際に動作するかどうかは，実行するまで確認できない。AutoView で作成されるアプリケーションのコールバックは，あらかじめ用意されている関数群と，属性の受け渡しによって自動的に作成されるコールバック，そしてユーザが独自に用意するコールバックと 3 種ある。後者の 2 つはその都度作成されるものであり，エラーを発生する可能性が高い。

エラーの主な原因は以下のものである。

- 属性の受け渡し時の変数の型が相違。
- 存在しない部品の呼び出し。
- ポインタ等のエラーが検出できない文法ミス。

これらのエラーの発生原因を，ユーザがデバッカを用いて，エラーの原因を探しそれを修正するのは大きな負担となる。そこで，AutoView ではアプリケーション実行時に発生したエラーをユーザに通知するシステムを導入している。

4.3 ヘルプシステム

オンラインヘルプ

アプリケーションを作成中に AutoView の使用方法や，作業の流れなどが判らなくなる場合がある。ユーザは AutoView の，画面に現われているものに対して"Help" キーを押すことによりその状態や，操作方法について知ることができる。

ソースファイルに対するヘルプ

AutoView によって作成された GUI およびコールバックは，最終的に C のソースコードとして展開される。そのソースファイル中には AutoView がコメントとして自動的に

ヘルプメッセージを挿入する。これにより，生成されたソースに対して，ユーザが変更を加える場合には作業を容易にすることができる。

5 今後の課題

今後さらに以下の機能を実装し，総合的なエンドユーザのアプリケーションの作成支援を行なう予定である。

- 対話技法の変更
  同等な機能をもつ対話技法の変換。例えばリストからプルダウンメニューなどへの動的変更。
- 自動的なレイアウト作成
  ユーザが部品の作成位置の細かい指定を行うのではなく，ある程度の範囲で部品の位置の整列を行う。
- 複雑な対話制御の作成手法
  多入力，多出力を行なうことができる対話制御支援。

6 おわりに

本稿で述べた機能をエンドユーザに提供することにより，GUIプログラミングの知識を持たないユーザでもGUI, 対話制御の作成が行なえると考えられる。今後は機能の充実と，このアプリケーションの有効性について検証する予定である。

参考文献

[1] 守屋 慎次，中谷 吉久. コンピュータの対話機構と直接操作インタフェース. システム制御情報学会誌, Vol. 33, No. 11, pp. 568–575, 1989.

[2] 太田原 剛，杉山 高弘. 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, 5–101–5–102 ページ, 1992.

[3] 吉山 正治，笠原 宏. Autoviewのコールバック作成負担軽減機能. 第 *45* 回全国大会講演論文集, 201–202 ページ, 1992.

[4] 吉山 正治，笠原 宏. AutoView:エンドユーザを対象としたUI構築ツール. 電気学会 電子・情報・システム部門第2回大会,117–118 ページ, 1992.

[5] NeXT Inc. *Interface Builder*. NeXT Inc, Palo Alto,CA, 1991.

[6] Dan Heller. *XView Programming Manual*. O'Reilly & Associates,Inc., 1990.