

3 K-10 知的システム構築用シェル AUK の構築支援環境 (3) グラフィカルユーザインタフェース構築環境

○桑原修二

藤村茂

富田昭司

横河電機株式会社

1 はじめに

現在、グラフィカルなユーザインタフェース(以下 UI)の構築用のシステムの開発が盛んに行なわれている。エキスパートシステム構築用ツールにおいても、作成されるアプリケーションの UI 構築用や知識ベースの開発環境の UI 構築用として UI 構築用のシステムが利用されている。それによって開発効率や保守性の向上をねらっている。

本稿では、まず Smalltalk-80 上で開発した汎用の UI 構築用カードツール Talkie について紹介する。次に、知的システム構築用シェル AUK[1]において実現した、AUK の知識表現、推論機構および構築方法論に基づく知的システムの UI 構築環境について述べる。

2 Smalltalk-80 上のユーザインタフェース構築用カードツール Talkie

2.1 ユーザインタフェースの情報を保持するオブジェクト G を組入れた MVC

ダイレクトマニピュレーションを活かした UI では、その状態やグラフィカルな属性など UI 自体に関する情報の重要性が増す。Smalltalk-80 の MVC(Model-View-Controller)においてグラフィカルな情報を保持するオブジェクトである View は、例えば、消してしまえばその画面上の配置などを再現することができないという点で、一時的なものであり、UI 自体の情報を永続的に保持する機能は十分でない。

UI の情報を保持するために、その情報を Model に付加することが考えられる。しかし、この方法では既存のアプリケーションを適用したい場合、Model が UI の情報を持っているために、アプリケーションは Model のサブクラスとして作ることを強制されてしまう。

そこで我々は、UI(View-Controller)とアプリケーション(Model)を表す MVC に、UI に関する情報を保持する

ためのオブジェクト G(Graphical model)を導入した。G の持つ情報は識別子、グラフィカルな情報(スーパービュー上での領域、境界の幅、色、内部の色、表示される絵、文字)、スクリプト、アプリケーションの情報(実体を得るためのメソッドや情報取り出し、設定のためのメソッドのメッセージセクタ)自分の上に貼りつけられているグラフィカルなオブジェクトの G などである。

G はユーザがグラフィカルなオブジェクトをグラフィカルな環境で作る時に生成され、View と Controller の変数 gModel に入れられる。また、G の依存性の辞書 DependentsFields(Smalltalk-80 の変更更新通知のための辞書)に View を登録する。

一方、アプリケーション(Model)は、何の変更も加えることなしに、ユーザが好きな時にグラフィカルな環境の下で設定できるようにする。内部的には、G のアプリケーションの情報が設定され、G はこれらの情報を依存性の辞書を介して自分の View に伝える。つまり、従来の Pluggable MVC[2]と同様に View-Controller にプラグインする形で Model が設定される。

G により UI の情報を保持でき、Smalltalk-80 の変更更新通知の仕組みにより MVC および G の互いの変化を反映でき、従来の MVC にも沿う形でアプリケーションに何の変更も加えることなしに、動的に設定できる。

2.2 Talkie におけるユーザインタフェース構築法

Talkie は 2.1 で述べた MVC および G のグラフィカルなオブジェクトから成る Smalltalk-80 上の UI 構築用のカードツールである。Talkie を用いた UI の構築法について述べる。

オブジェクトにはカード、ボタン、フィールド、リスト、スクリプト、コンテナ等があり、種類、構成は Macintosh の HyperCard[3]にほぼ従ったものとなっている。1つのまとまった情報はカードとして表現する。カード上では、ボタン、フィールド等の動的な生成、削除、属性の変更が行なえる。図 1 に実行中の Talkie の画面を示す。

カード上のオブジェクトに対する操作は、従来の

Smalltalk-80での操作法[4]に従うとともに、ダイレクトマニピュレーションを実現している。HyperCard などの操作と同様のものである。また、ユーザの用途に応じてシステムの状態を変えるユーザレベルを導入している。

まず、ユーザレベルをauthor(UI作成モード)にして、新しいカードを作る。なお、作られたオブジェクトがカードの場合、そのGはコンテナという辞書に登録される。そのカードを再び開く時は、コンテナの情報をもとに開かれる。開いたカードの上にボタン、フィールドなどを貼りつけ、自由に配置、グラフィカルな属性の変更をしていく。また、ボタンのスクリプト(起動されるSmalltalk-80のソースプログラム)の記述やカードへのリンクを張ったりする。なお、カード上に貼りつけたオブジェクトのGはカードのGに保持される。

ここまでの操作では、View-Controller および G が存在する状態であり Model は設定されていない。しかし、アプリケーションを起動するためのフロントエンドやオーサリングシステムとしてならばこのままでも使用できる。

さらに、Model(アプリケーション)を設定したい場合は、設定用のカード(ダイアログカード)を開き、Modelの実体(を得るためのメソッド)や Model に送ればいい数個のメソッドのセレクトタなどを記述し、accept する(Gに登録する)。G に設定されたこれらの情報は、依存性の辞書を介して、自分の View にも反映される。

ユーザレベルを browser(使用モード)にすると、ユーザの操作に応じて登録したアプリケーションのセレクトタのメソッドが起動される。

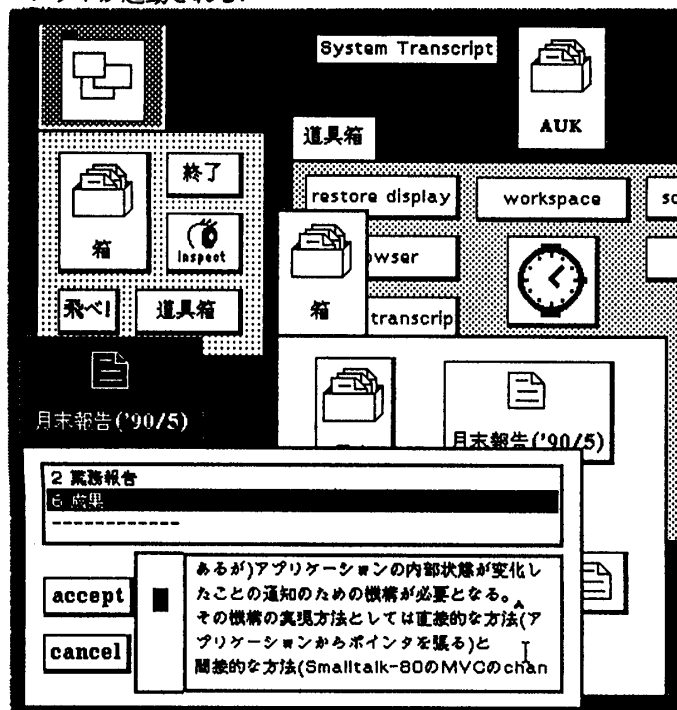


図1 Talkie の使用例

3 AUK におけるユーザインタフェース構築環境

知的システムにおけるユーザとシステムとの相互の情報授受としては以下の4つが考えられる。

1. システムが必要時に入力をユーザに催促し、入力が終了すると制御をシステムに戻す。
2. システムが必要時に出力を行い、終了すると制御をシステムに戻す。
3. ユーザが必要時に入力を行う。
4. ユーザが必要時に出力を要求する。

AUK においては、ユーザも一つの知識モジュールとして考え、ユーザに対する入出力は、aukAccessor の一種の userAuk に対して行うものとしている。こうすることにより、トップダウン設計を行う場合、ユーザに対する入出力を意識せずに統一的にモジュール仕様を決定できる。1,2 は入出力用のカード群を扱う機構を付加した userAuk として実現することにする。

3,4 は、Talkie においては Smalltalk-80 の変化更新の通知機能によって UI とアプリケーションを分離しつつ実現可能である。あるカード上のボタンを押すことによってユーザが必要に応じて開くことができるようにして用いる。

なお、Talkie のカードは、ダイアログカードを用いて Model の情報を設定する必要がある。Model が aukAtom(auk の基本要素)の場合は、スロット名の指定のみでこれらが指定できるようにしている。

4 まとめ

Smalltalk-80 上で UI 構築用カードツール Talkie を開発した。その上に知的システムの UI の構築と知識ベースの構築とを統合した知的システム構築環境を開発した。

参考文献

- [1] 藤村他:知的システム構築用シェル AUK の構築支援環境 (1) その設計概念, 情報処理学会第 41 回全国大会 (1990)
- [2] 青木淳:使わないと損をする Model-View-Controller, 富士ゼロックス情報システム資料 (1988)
- [3] Goodman, D.: *The Complete HyperCard Handbook*, Bantam Books (1987)
- [4] Goldberg, A.: *Smalltalk-80, The Interactive Programming Environment*, Addison Wesley (1984)