

DEVO: GUIプレゼンテーション・フレームワーク

増田 英孝, 菊池 肇, 笠原 宏 (東京電機大学 工学部)

概要

筆者等は、グラフィカルユーザインタフェース (GUI) を容易に実現するための統合的な支援環境を構築するために UIMS (User Interface Management System) の研究を行なっている。

本稿では、UIMS のプレゼンテーション・フレームワークとして DEVO (Dynamically Extended View Objects) を提案する。DEVO はダイナミックな取り扱いができるビューであり、UIMS の表示定義における GUI レイアウトの編集支援、アプリケーション実行時に GUI のプレゼンテーション変更を支援するために利用できる。

DEVO: A GUI Presentation Framework

Hidetaka MASUDA, Hajime KIKUCHI, Hiroshi KASAHARA
Department of Electrical Engineering, Faculty of Engineering
Tokyo Denki University,
2-2 Kandnishiki-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.

We propose DEVO (Dynamically Extended View Objects) as a new UIMS presentation framework. DEVO is dynamically manageable and it is useful not only for assisting GUI layout editing during the UIMS presentation definition, but also for assisting GUI presentation modifications after starting up applications.

1 はじめに

筆者等は、GUI (Graphical User Interface) を容易に実現するための統合的な支援環境である UIMS (User Interface Management System) を研究テーマとしている。UIMS は、アプリケーションの GUI 部を効率よく構築するライブラリとその構築を支援するツールから構成される [1]。

現在、Objectworks\Smalltalk Release4.0 [2] [3] (以後、Smalltalk と表す) 上でこの環境を構築しており、初心者にも利用可能な、GUI の対話的・視覚的な作成・編集環境を構築することを目的としている。

本研究では、動的な取り扱いができるビュー - DEVO (Dynamically Extended View Objects) [4] を提案し、GUI プレゼンテーション部の支援を行なっている。DEVO は、ビューのレイアウト変更、ビュー階層の変更を可能とする。このため、DEVO は、GUI 表示定義におけるレイアウト編集支援、アプリケーション起動後に GUI の変更を支援するために利用可能である。また、DEVO の機能はいつでも呼び出しが可能であり、ユーザが対話的に利用可能である。

本稿では、DEVO の GUI プレゼンテーション部支援機能について述べる。

2 UIMS としての Smalltalk

UIMS は、アプリケーションの GUI 部を効率よく構築するライブラリとその構築を支援するツールから構成される [1]。

Smalltalk は言語だけではなく、アプリケーションの実行環境をも含み、一種の UIMS とみなすことができる。また、GUI アプリケーション作成のためのパラダイムとして MVC モデル [3] が用意され、多数のクラスライブラリが提供されている。

- GUI 構築のためのパラダイム

MVC (Model-View-Controller) モデル

- GUI 部を効率よく構築するライブラリ MVC のクラスライブラリ
- GUI 構築を支援するツールプログラミング
- UI ガイドラインプログラマに一任

しかし、GUI 構築を支援するツール、明確な GUI ガイドラインは用意されていない。このため、GUI 構築を支援するツールおよび GUI ガイドラインの作成が望まれる。

3 Smalltalk のビュー構成

図 1 に一般的な Smalltalk のビューを示す。Smalltalk ではウィンドウやウィンドウを構成する GUI 部品をビュー (View) と呼ぶ [2]。図 1 のように複数のビューを組み合わせ、複合オブジェクトとして一つのウィンドウを構成している。

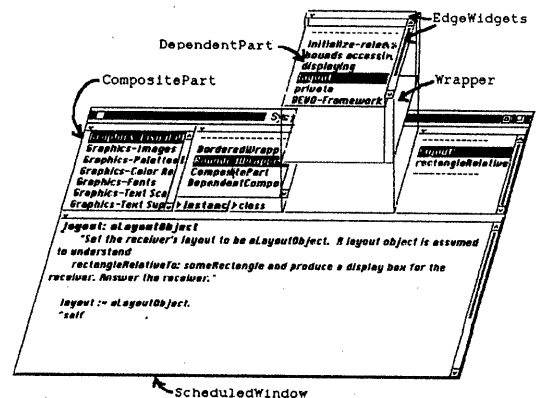


図 1: Smalltalk のビュー構成例

図 2 にビュー階層の例を示す [3]。Smalltalk のウィンドウも、他の GUI ツールキット同様にインスタンスの階層を持つ。

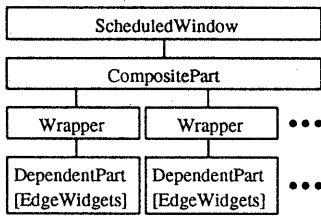


図 2: Smalltalk のビュー階層例

ScheduledWindow のインスタンスがビューの土台となり、ウィンドウのタイトルを付けたり、ウィンドウに対する操作を取り扱う。

その ScheduledWindow のインスタンス上に CompositePart のインスタンスがある。CompositePart は複数のビューをとりまとめるコンテナの役割を果たす。

更にウィンドウの部品を表す DependentPart (またはサブクラス) のインスタンスが複数存在する。

Wrapper は、具体的な部品である DependentPart を包み込む。更に、DependentPart と部品の装飾を行なう EdgeWidget を包み込んで一つの部品としてまとめることもできる。Wrapper で包まれた DependentPart と EdgeWidget は一つの意味のある部品として利用することができる。例えば、メニューバーとスクロールバーを持つテキストやリストがある。DependentPart は組み合わせられるデフォルトの Wrapper クラスを持つ。

これらのビューは通常、アプリケーション起動時から終了時まで表示領域や構成が変更されることはない静的なビューであり、動的な取り扱いは考えられていない。

4 DEVO (Dynamically Extended View Objects)

DEVO は動的にビューを取り扱うことができるフレームワークと、その操作環境を含む。

DEVO の基本機能として、

- レイアウトの変更
移動、リサイズ
- ビュー階層の変更
グループ化、アングループ化
- レイアウトの整形
グループ内のビューの整形

がある。これらの基本機能を組み合わせて、更に複雑な機能を実現できる。図 3、図 4 に DEVO の基本機能を示す。ウィンドウ内の GUI 部品 (ビュー) に対して、移動、リサイズ、グループ化を行なうことができる。

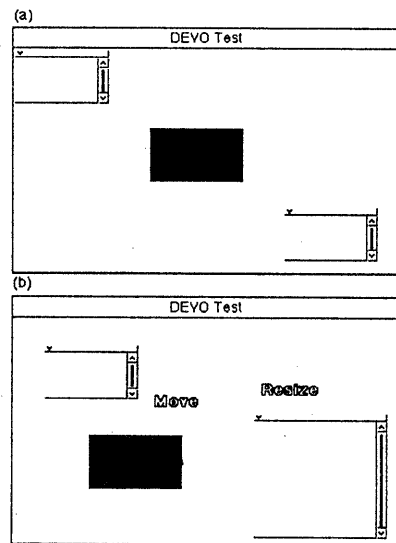


図 3: DEVO の基本機能 1

図 3 は、GUI 部品に対して、移動、リサイズを行なった例である。GUI 部品は既存の部品をそのまま利用することができる。Wrapper に対してメッセージを送信することによって DEVO の機能呼び出す。

図 4 は、リストとボタン群をグループ化した例である。グループ化によって、複数の部品を

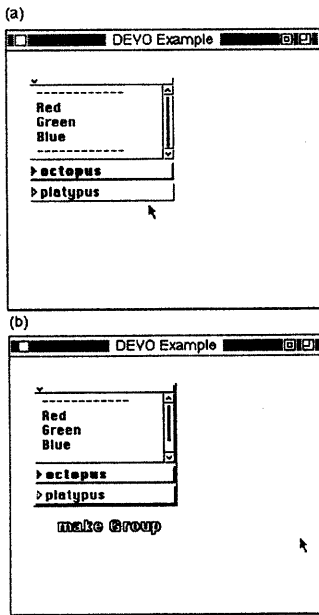


図 4: DEVO の基本機能 2

まとめて取り扱うことができ、枠をつけることができる。同一階層内の複数の Wrapper を一つの CompositePart で取りまとめてグループを実現している。また、グループ化したものをアングループ化することも可能である。

Smalltalk では、配置に関する情報が Wrapper に切り出されている。本研究ではその点に着目し、Wrapper のインスタンス変数 layout を変更し、ビューのレイアウト変更を可能にしている。

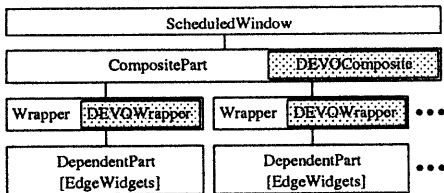


図 5: DEVO Framework

図 5 に DEVO Framework の構成を示す。レイアウトの変更機能を DEVOWrapper が、ビュー階層の変更およびレイアウト整形機能を DEVOCComposite の部分が受け持つ。

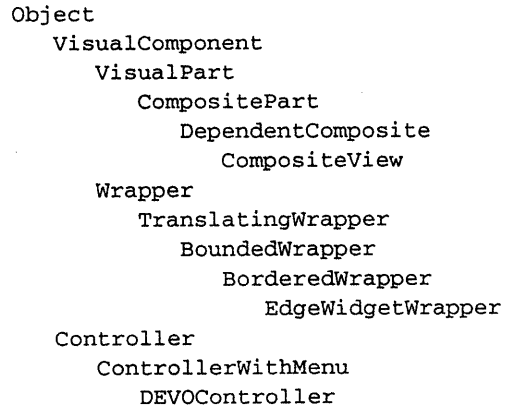


図 6: DEVO の関係するビュークラス階層

DEVO の機能は、図 6 に示したクラス階層中の BoundedWrapper 以下のサブクラスのインスタンスすべてに利用可能である。結果として、BoundedWrapper 以下のインスタンスすべてに貼り付く形になる。そのため、既存のアプリケーションであっても、変更することなく DEVO の機能をそのまま利用可能である。また、ユーザにとって、DEVO が起動されなければ通常の Smalltalk 環境と等しい。

DEVO の機能を利用するには、DEVOCController を利用するか、直接 Wrapper にメッセージを送ればよい。

4.1 DEVOWrapper

BoundedWrapper には、インスタンス変数として layout があり、Rectangle もしくは LayoutFrame のインスタンスが保持されている。この layout に基づいて Wrapper の配置が決定され表示領域が求められる。また、ウィ

ンドウ全体のリサイズの際も layout に基づいて描画位置が再計算される。

DEVO では、move: aPoint, resize: aRectangle メソッドを用意し、メッセージを送信することによって、移動、リサイズを行なうことができる。

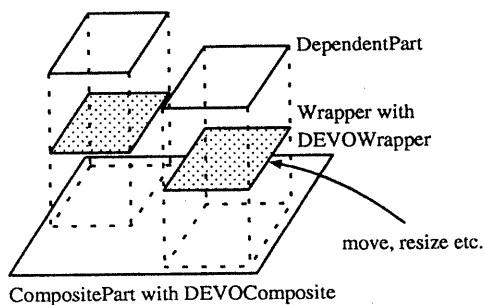


図 7: DEVOWrapper の構成

図 7に示すように、DEVOWrapper は、親の CompositePart 内部を自由に移動、リサイズすることができ、Wrapper の layout にその変更が反映される。

4.2 DEVOWrapper

DEVOWrapper は、複数の Wrapper の配置を管理する。また、同一階層内の複数の Wrapper をグループ化したり、自分をアングループ化する機能を持つ。配置のルールを変更することによって、CompositePart 内の Wrapper の配置を変更することができる。

Smalltalk では、layout として LayoutFrame を利用することができる。LayoutFrame は、部品にオフセットを指定することにより、相対位置で部品を配置することができる。例えば、ウィンドウのリサイズの際に、ボタンの高さをテキストの高さに固定し、ボタンの上や下の部品のサイズ変更で領域を調整するために使われる。

同一の階層内で LayoutFrame を多用すると、部品の配置がわかりにくくなる。複数の関連する部品はグループとしてまとめて管理すべきである。配置の変更もグループ内やグループ単位で行なうべきである。

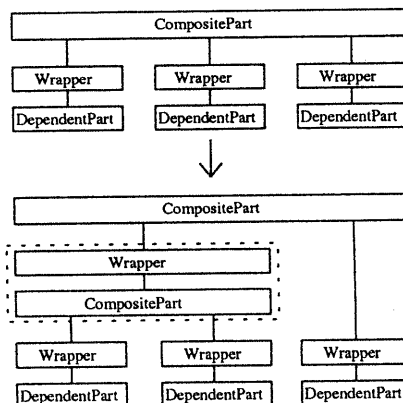


図 8: グループ化

図 8にグループ化の際のビュー階層の変更を示す。ビュー階層の間にグループとして利用するビューを挿入することができる。このグループ化によって、構成要素グループに枠をつけて明示でき、レイアウト変更の際に細かくブロック化でき、再利用の単位としても利用できる。

4.3 DEVOWrapper

ControllerWithMenu のサブクラスとして、DEVOWrapper を用意している。DEVOWrapper はユーザからの入力を受け取り、DEVO Framework に指示を出すユーザインタフェースの部分を受け持つ。DEVO に対する操作をメニューで提示し、ユーザからの入力を処理する。

ユーザが対話的に DEVO の機能を利用する場合には、図 9に示すように、ビューのコントローラの代わりに DEVOWrapper を利用

5.1 GUI 表示定義におけるレイアウト支援

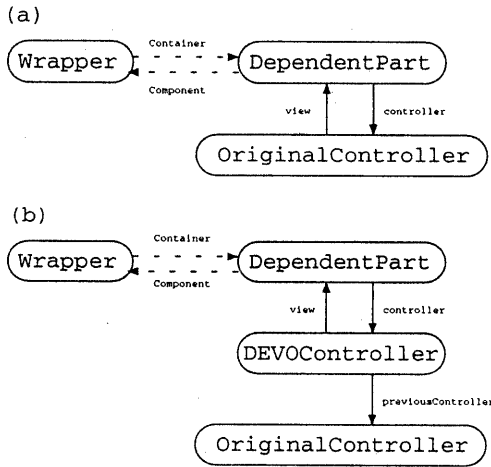


図 9: DEVOController (a) 通常の状態 (b) DEVOController 使用状態

する。DEVOController は、ビューの元のコントローラを保持する。そのため、アプリケーションが動いている時にもコントローラを取り替えることによって DEVO の機能を利用することができる。ウィンドウ上の GUI 部品を選択し、指示を出す機構を DEVOController が受け持つ。ビューの単一選択、複数選択を用意しているため、任意の階層のビューを選択することができる。

5 GUIプレゼンテーション部への適用

GUIプレゼンテーション・フレームワークとして、GUIの表示定義ツールおよび、アプリケーション起動後にGUIのレイアウト変更を行なうために DEVO を利用することができる。以下に作成中の GUI 構築ツールとアプリケーション起動後に GUI レイアウトを変更可能とする環境の概要を示す。

視覚的・対話的な GUI 構築ツールを作成する際に、GUI 部品のレイアウト機能は必要不可欠である。DEVO の基本機能を利用することによって、GUI 部品をレイアウトする機構部が容易に実現できる。また、コントローラを DEVOController と置き換えることによって、既存の GUI 部品 (ビュー) を GUI 構築ツールでそのまま利用することができる。

筆者等は、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っているが、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 構築ツールの開発を行なっている [5]。その一つとして OSF/Motif を対象とした GUI 構築ツール MANU がある [5]。現在、Smalltalk 上に、GUI の表示定義ツールとして MANU の対話方式に沿ったツールを構築中であり、DEVO を利用している。サンプルの提示、属性の変更、グループ化に基づく GUI 作成の対話方式をサポートする。図 10 にツールのユーザインタフェースを示す。

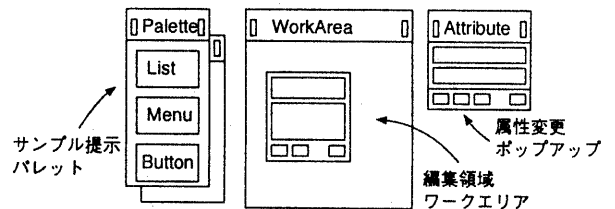


図 10: GUI 構築ツールの UI 構成

ツールはサンプルを提示するパレット、ユーザが作業を行なうワークエリア、GUI 部品やグループの属性を設定する属性変更ポップアップから構成される。DEVO の機能を利用して、既存のアプリケーションの GUI を再利用する機能も実現可能であり、現在実装中である。

5.2 アプリケーション実行時 GUI 変更支援

既存のアプリケーションの起動後に、GUIの変更を行なうために DEVO を利用できる。アプリケーション実行時に DEVO の機能を利用し、既存のアプリケーションに手を加えることなく GUI の変更を行なうことができる。

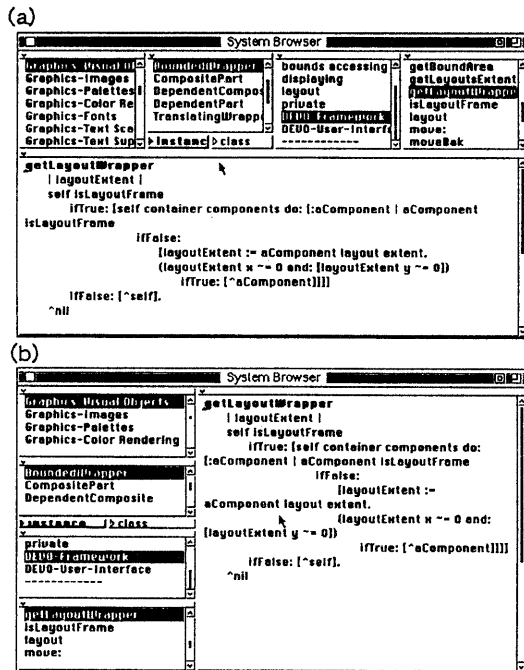


図 11: AP 実行時レイアウト変更 (a) 変更前 (b) 変更後

通常、アプリケーションのユーザインタフェースはアプリケーション作成時から固定されており、そのスタイルが変更されることはない。最近のシステムでは、ユーザが GUI をカスタマイズすることが可能であるが、アプリケーションを起動する前にその設定を行なわなければならない。また、設定方法は文字ベースであり、更に設定の変更には思考錯誤が必要であるため、設定・実行の繰り返しが必要になる。

そして、同一のユーザであっても、アプリケーションの習熟度が進むことによって適するユーザインタフェースは異なる。アプリケーション起動後に、ユーザが GUI のカスタマイズを視覚的に変更できるべきであるが、このような機能は従来のシステムでは実現されていない。

DEVO を利用することによって、GUI 作成時だけでなく、既存のアプリケーションを起動後に GUI の変更を行なうこともできる。DEVO のモードに入ることにより、コントローラが DEVOController に置き換えられ、DEVO の機能を利用することができる。

図 11 にアプリケーション起動後に GUI レイアウトを変更した例を示す。これは Smalltalk に対しても GUI 部品のレイアウトを変更した例である。SystemBrowser は対話の流れが左から右に流れるスタイルであるが、レイアウト変更によって上から下に対話の流れるスタイルに変更している。

6 実装予定の機能

今後、DEVO 上に実装予定の機能として、

- コンストレイントを伴ったレイアウト変更機能
個別にビューの配置を変更するだけでは不十分であるため、ビューの配置変更に対して表示上関連するビューの領域を自動的に調整する機能が必要である。
- ユーザのログ記録
ユーザがどのような場合にどのような操作をしたかを記録し、後からユーザの作業の分析を可能にする。
- ファジィ推論を用いたレイアウト自動整形機能 [6]
グループ化したビュー内の GUI 部品を自動的に整形する機能 [5]。

- GUI 部品の置換
機能的に置換可能な GUI 部品を変更する機能 [5]。
- グラファを容易に実現する機能 [7]
対話制御を記述する Statechart エディタを構築するために、グラファを容易に編集可能なビューを実現する。

がある。これらの機能を実現し、GUI の作成、変更を支援する環境を構築していく予定である。

7 おわりに

本稿では、動的な取り扱いができるビュー DEVO を提案し、GUI プレゼンテーション部の支援について述べた。DEVO は、ビューのレイアウト変更、ビュー階層の変更を可能とする。このため、DEVO は、GUI 表示定義におけるレイアウト編集支援、アプリケーション起動後に GUI の変更を支援するために利用可能である。

今後、対話制御部の支援も行ない、GUI の詳しい知識を持たないユーザにも容易に作成・変更可能な GUI カスタマイズ環境を構築する予定である。

最後に、Smalltalk に関する筆者等の質問に丁寧に答えて頂いた株式会社 SRA ソフトウェア工学研究所の松岡泰史氏、渡邊克宏氏に感謝致します。

参考文献

- [1] 宮崎: ユーザインタフェース管理システムと対話制御, 情報処理, Vol.33, No.11, pp.1295-1303 (1992).
- [2] 富士ゼロックス情報システム株式会社: *Objectworks\Smalltalk Release4 User's Guide* (1990).
- [3] 青木: *Smalltalk* ソフトウェア開発, SuperASCII, Vol.2, No.6 - Vol.3, No.5 (1991-1992).
- [4] 増田, 笠原: *DEVO (Dynamically Extended View Objects): UIMS のためのフレームワーク*, 情報処理学会第 45 回全国大会, 2T-5 (1992).
- [5] 増田, 大谷, 中岡, 小塚, 宮崎: *MANU (Motif Architect for Nonexpert Users): エンドユーザを対象とした GUI 構築ツール*, 情報処理学会ヒューマンインタフェース研究会, 44-17 (1992).
- [6] 和田, 笠原: 人のファジィ的な感覚に合うメソッドの決定を行なう *Smalltalk-80* とその応用, 情報処理学会第 45 回全国大会, 5U-8 (1992).
- [7] 増田, 笠原: *Smalltalk-80* における拡張 MVC モデルとその応用, 情報処理学会論文誌, Vol.31, No.2, pp.259-267 (1990).