

U I 開発ツール U-face II

- マルチウィンドウアプリケーション 開発への展開 -

神場 知成*, 井関 治**, 久松 欣一***

* 日本電気(株) C & C システム研究所

** 日本電気(株) 関西 C & C 研究所

*** 日本電気テレコムシステム(株)

筆者らの提案するユーザインタフェース(U I)設計法(マルチビューモデル)に基づくU I 開発ツール U-face II について述べる。筆者らは従来から、ホスト接続型端末を設計対象とするU I 開発ツール(U-face)をワークステーション上に開発してきたが、今回それをパソコンのOS/2プレゼンテーションマネージャ上に移植、拡張し(U-face II)、ウィンドウ設計機能、ソースコード出力機能などを強化した。U-face IIにより、U I 開発者はプログラミングをすることなく、マルチウィンドウU Iの開発が可能になる。

U-face II: UI Building Tool

- A Step to the Development of Multi-window Applications -

Tomonari KANBA*, Osamu ISEKI**, Kin-ichi HISAMATSU***

* C&C Systems Research Laboratories, NEC Corporation
1-1, Miyazaki 4-chome, Miyamae-ku, Kawasaki-shi,
Kanagawa 216 Japan

** Kansai C&C Laboratory, NEC Corporation

*** NEC Telecom Systems Limited

This paper describes a user interface (UI) building tool, U-face II. This system is based on the multiview model, which is a UI design methodology proposed by the authors. The design targets of the former version of U-face II, which was called U-face, were host-connected terminals. The design targets of U-face II are multi-window systems. U-face II is implemented on the Presentation Manager of OS/2, and it has window design function, source codes output function, etc. U-face II allows the UI designers to build multi-window UIs without describing programs.

1. はじめに

ディスプレイ画面上でのユーザインタフェース（UI）が複雑化するにつれ、その設計サポートの研究が盛んになっている。それらは、手法毎にそれぞれ異なった特徴を持つ。例えば、設計用のツールキットをプログラムライブラリの形式で提供する手法は、UI設計ではなく実装の段階で大きな効果を持つ。ビジュアルプログラミングは、ノンプログラマによるUI設計を可能にする。

これらの手法は、それぞれUI実装の効率化や、UI設計者の層の拡大などに大きな効果を持つが、それらを用いて設計したUIの品質は、必ずしも保証されていない。これに対し筆者らは、高品質なUIの設計をサポートするという立場から、マルチビューモデルと呼ぶ考え方にに基づくUI開発ツール U-faceを開発してきた¹⁾²⁾。U-face は最初、ホスト接続型端末等、シングルウィンドウのUIを設計対象としてワークステーション上に開発したが、今回これをパソコン上に移植、拡張するに当たり、システム構成を変更し、マルチウィンドウを利用した柔軟なUIを開発可能なものとするに決めた。本稿では、これまで開発してきた U-face の概要と、その開発コンセプトを継承してパソコン上に構築中のU-faceⅡについて述べる。

2. U-face

2.1 マルチビューモデル

一般に、高品質なUI開発には、プロトタイプ作成とその評価を繰り返す、いわゆる繰り返し設計が有効であると言われている。しかし従来のプロトタイピングツールは、作成したUIの評価が人間の主観に依存しているため、インタラクションが複雑なUIでは、設計した機能全体の把握、その問題点の抽出が非常に困難であった。

この問題に対処する方法として、筆者らが開発した U-faceでは、マルチビューモデルと呼ぶ考え方をとっている¹⁾。一般にユーザがシステムに接するときには、システムを多数の側面から見る。また、それらは必ずしも一貫していない。例えば、あるユーザは画面のデザインを重視するであろうし、別のユーザは操作が少ない手順でできることを重視するであろう。同一のユーザが時によって異なる見方をすることもある。マルチビューモデルは次のような考え方にに基づくUI開発法である。

- UIは複数の側面を持ち、それぞれの側面毎に複数の記述法が存在する。側面と記述法の両方を限定することによって決定する1つの記述をビューと呼ぶ。

- UI設計においては、設計中のUIを、設計中のビューとは別のビューに変換、表示してUIの種々の側面を検討しながら進めるべきである。

2.2 U-face の概要

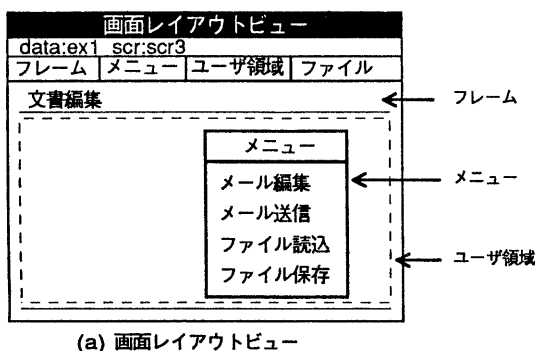
U-face は、マルチビューモデルに基づく、ホスト接続型端末画面のUI開発ツールであり、NEC製 EWS4800の X-Window上で開発されている。マルチビューモデルにおけるビューの表示、変換等の機能が実現されている。

U-face では、複数あるビューの中で、設計に利用するものを設計ビュー、評価に利用するものを検証ビューと呼んでいる。それぞれの概要を以下に示す。

1) 設計ビュー

- 画面レイアウトビュー（図1(a)）

画面の外見を設計するビューである。背景、メニュー等、画面の構成部品を描画ツールで設計し、それらを組み合わせて画面レイアウトとして登録する。



カーソル位置	入力	応答
menu1	↓	カーソルを次項目
menu1	↑	カーソルを前項目
menu1	M_CLICK	カーソルを指定項目
menu1:01	RET	画面を<scr2>に移動

(b) 操作規則ビュー

図1. U-face の設計ビュー

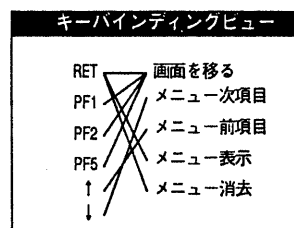
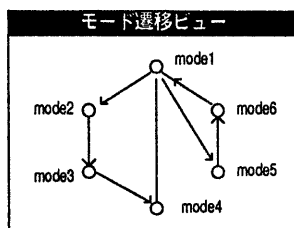
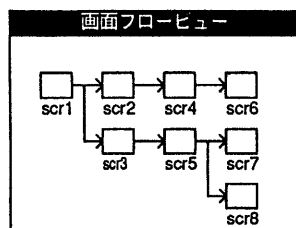


図2. U-face の検証ビュー

• 操作規則ビュー (図1(b))

各画面におけるユーザ入力とシステム応答との対応関係を設計するビューである。画面毎に操作規則を表形式で記述する。

2) 検証ビュー

• シミュレーションビュー

このビューは、設計したUIの動作を、実際に利用される場合と同じようにステップ毎に確認するためのものである。ユーザがキー押下、マウスクリックなどの操作を行うと、それに対応するシステム応答が操作規則ビューで作成された表から検索、実行される。

• 画面フロービュー (図2(a))

このビューは、バランスのとれた画面フローの作成をサポートする。作成した画面のうちの1つを指定すると、その画面を起点とする画面フローをダイアグラムで表示する。メニューにおいて選択肢が多すぎたり、1つの機能を実行するまでの画面遷移が多すぎたり、という事態を避けることができる。

• モード遷移ビュー (図2(b))

このビューは、理解しやすいモード遷移の作成をサポートする。操作規則表に基づいて、英数字入力、仮名入力、メニュー選択などのモードを導いて、それらのモードの遷移関係をダイアグラムで表示する。

• キーバインディングビュー (図2(c))

このビューは、一貫性のあるキーバインディングの設計をサポートする。設計したキーとその意味づけとの対応関係をグラフで表示するので、場面によって異なった意味づけのされたキーをUI設計者は容易に発見できる。

U-face は、いわゆる「使い捨て型のプロトタイピングツール」であり、設計したものを実装する際には新たなコーディング等が必要である。この手法は、UI品質の向上に効果を持つことは認められているが⁶⁾、実装まで含めたUI開発効率の向上という点では十分ではない。

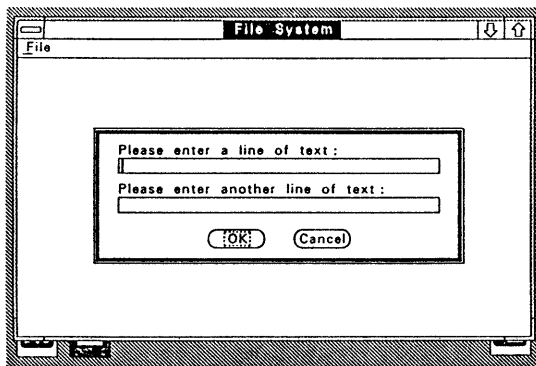


図3. ダイアログボックス (文献(4)より引用)

3. U-faceⅡ—マルチウィンドウ画面のUI設計

3.1 概要

U-faceⅡはOS/2のプレゼンテーションマネージャ(PM)上のUI開発ツールであり、NEC製のPC-9800系マシン上で開発中である。基本的なコンセプトはU-faceを踏襲し、設計ビューと検証ビューとから成るが、設計対象をPM上のマルチウィンドウアプリケーションとするに当たり、システム設計に全面的な変更を行っている。主な変更点には次のものがある。

- 設計ビューにおいては、ウィンドウシステム上の各ウィンドウ、その表示内容、操作手順の設計が可能である。検証ビューも、ウィンドウシステムのUI検証に対応している。
- 設計結果はCのソースコードとして出力され、そのまま実装することができる。

3.2 設計ビュー

1) 概要

OS/2はPM上のUIを実現するためのグラフィクスオブジェクトとして、アイコン、ダイアログボックスなどを持っており、それらオブジェクトのビジュアルな構築をサポートするエディタを提供している。例えばダイアログボックスエディタは、ボタン、スクロ

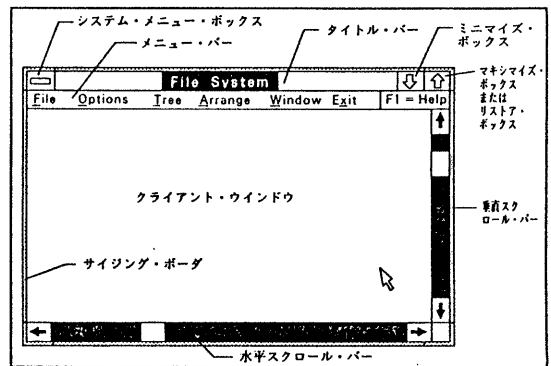


図4. スタンダードウィンドウ
(文献(4)より引用)

ールバーなどの部品を配置して、オブジェクトを作成できる(図3)。

しかしPMでは、OS/2で最も頻繁に利用されるウィンドウであるスタンダードウィンドウ(図4)をビジュアルに設計するツールは提供されておらず、一般のアプリケーションのUI部分を設計するには複雑なプログラミングを必要とする。また、ビジュアルな設計ツールとして提供されているダイアログボックスエディタでは、ボックスの外見は設計可能であるが、操作手順に関してはプログラミングを必要とする。

U-faceⅡは、これらの機能を補い、アプリケーションのUI部分を可能な限りビジュアルに構成できるようにするために、スタンダードウィンドウの設計ツールを提供する。また、ウィンドウ上のメニュー選択やボタン操作に伴うシステムの応答は、あらかじめU-faceⅡが提供するコマンドを、各部品の属性として指定すれば良いので、操作手順のプログラミングが大幅に軽減される。

2) スタンダードウィンドウ

スタンダードウィンドウは、フレームウィンドウと呼ぶ土台上に、タイトルバーなどの部品と、アプリケーションに制御されるクラ

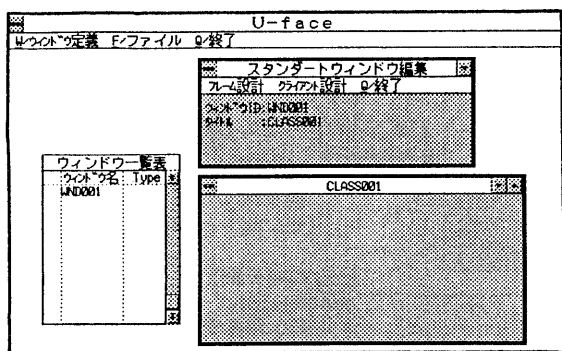
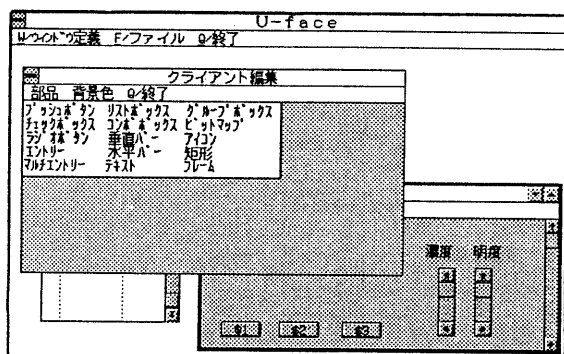
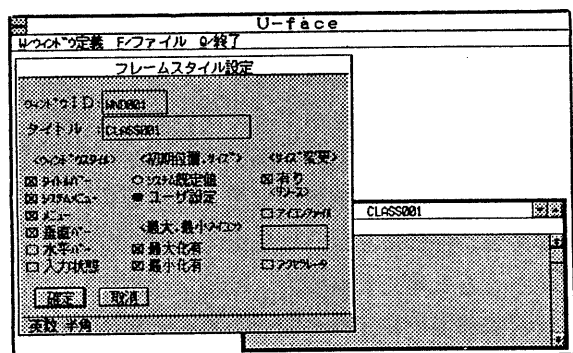


図5. スタンダードウィンドウ設計



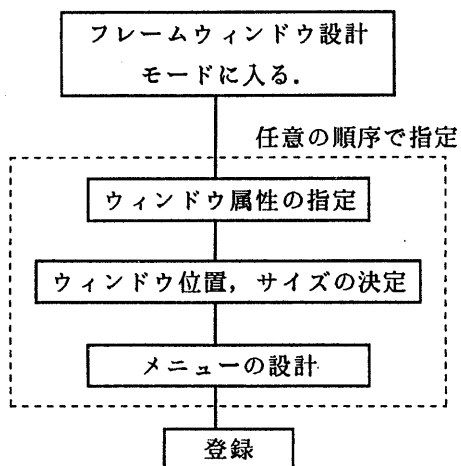
(a)クライアントウィンドウ設計の画面



(b)クライアントウィンドウ設計の手順

図7. クライアントウィンドウ設計

(a)フレームウィンドウ設計の画面



(b)フレームウィンドウ設計の手順

図6. フレームウィンドウ設計

クライアントウィンドウが表示されたものである。

U-faceⅡでは、これら2つの領域をモードを切り換えることによって設計し(図5)、設計結果はOS/2のコードとして出力する。これにより、AP側の機能が実現前であっても、UI側から呼ばれる名称に対応する疑似的な関数を用意すれば、UIの操作シミュレーションが可能である。

以下では、フレームウィンドウとクライアントウィンドウの設計方法について述べる。

3) フレームウィンドウ設計

フレームウィンドウ設計の画面と基本的な手順を図6(a)(b)に示す。

設計のプロセスの内容を次に示す。

a)ウィンドウ属性の指定

フレームウィンドウ内でのタイトルバー、

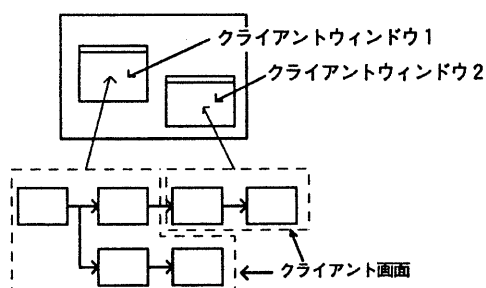


図8. クライアントウィンドウと
クライアント画面

メニュー、水平／垂直スクロールバーなどの表示の有無を選択する。選択結果は直ちに画面表示に反映する。

b)ウィンドウ位置、サイズの決定

表示されたウィンドウの初期位置、サイズをマウスを利用して設定する。

c)メニューの設計

メニュー設計用の表を用いてメニュー項目のデザインをすることが可能である。表は、表示文字列と識別名の欄から成り、各メニューが選択された時の実行機能も記述できる。

4)クライアントウィンドウ設計

クライアントウィンドウ設計の画面と、設計の基本的な手順を図7に示す。

クライアントウィンドウの設計機能は、次のような特徴を持つ。

a)各クライアントウィンドウに対して、1つ以上のクライアント画面が設計可能である(図8)。ウィンドウシステムにおけるアプリケーションとして、1つのウィンドウ内で画面が次々に変化するものと、1つのウィンドウには1つの画面しか表示されないカード型のものとがあるが、この機能により、いずれの型のものにも対応できる。

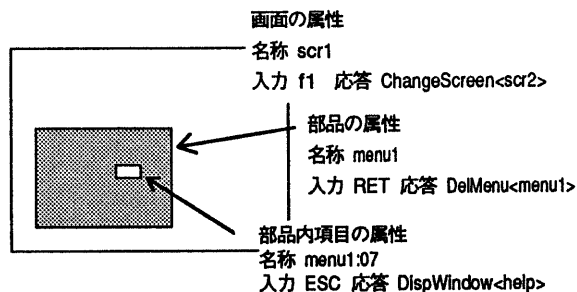


図9. 応答の定義

b)各クライアント画面内に、PM固有および独自開発の部品が配置できる。

c)部品には、ただ表示だけのもの(背景部品)と、ユーザの入力等を受理し、それに伴って表示が変化可能なもの(機能部品)とがある。機能部品は、背景部品に重ねて描画することも可能である。

- 背景部品：文字列や図形、イメージなど
- 機能部品：PM組み込みの各種ボタン、フィールド、垂直／水平バー、テキスト、ビットマップ、アイコンなどに、U-face IIでは、カスタムボタン(ビットマップデータとして作成した形状を、ボタンとして利用する機能)などの拡張を行う。また、データベース等のアプリケーションを想定し、アプリケーションが読み込んだデータを表示するキャンバスエリアを設定可能にする。

d)ユーザの入力とそれに対する応答は、画面に対して定義することも、部品に対して定義することも、部品内の特定の項目に対して定義することもできる(図9)。入力応答の定義が、上位のもの(画面、部品、部品内項目の順)と下位のもので重なった場合は、下位のもの優先する。

ただし応答欄には、機能名を直接記述するか、または、ツールの提供するメニュー項目

から選択入力する。以下の機能名を、複数個つなげて記述可能とする（引数省略）。

- DispWindow：ウィンドウを表示する。
- ChangeScreen：同一ウィンドウ内で、クライアント画面を変える。
- CloseWindow：ウィンドウを消去する。
- ActivateWindow：ウィンドウをアクティブにする。
- IconWindow：ウィンドウをアイコン化する。
- DispObject：部品を表示する。
- DelObject：部品を消去する。

この他の、アプリケーションの機能実行モジュールを実行するためには、それらの機能を外部関数としてコールする。

4) ソースコード出力機能

OS/2では、1つのアプリケーションを実現するために、アイコンのビットマップデータなどの他に、以下の4つのテキストファイルが必要である。

a) ソースファイル（拡張子 .c）

イベントを処理するメインループを構成し、ユーザ入力に対応する処理を行うプログラムの本体。

b) ヘッダファイル（拡張子 .h）

PMで使用する構造体やAPIファイル

c) モジュール定義ファイル（拡張子 .def）

プログラム名、ローカルヒープサイズ、スタックサイズなどリンクに知らせるべき情報。

d) リソーススクリプトファイル（拡張子 .rc）

ダイアログボックス、メニュー等の内容に関する情報。

U-faceⅡは、設計結果に基づきこれらすべての情報を出力する。

3.3 シミュレーション機能

U-faceⅡでは、部品から呼ばれるアプリケーションの外部関数に対して、疑似的な関数を割り当てておけば、操作シミュレーション

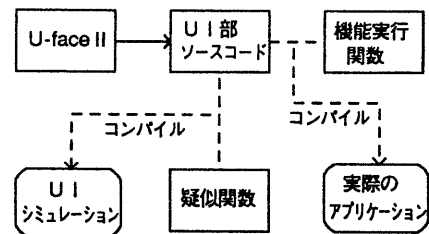


図10. U-faceⅡによるUIシミュレーション

が可能である。疑似的な関数を、実際の機能を持つ関数に置き換えることにより、アプリケーションが完成する。この機能により、アプリケーションの機能実行モジュールを実現する前にまずUI部を実装し、その操作イメージを確認しながらアプリケーション開発を進めることができる（図10）。

3.4 検証ビュー

U-faceⅡでは、3.2で述べた設計ビュー、および3.3で述べたシミュレーション機能に加え、UI評価および効率的な設計を目的とした以下の機能を開発中である。

1) ウィンドウ親子関係ダイアグラム作成機能

子ウィンドウは親ウィンドウに位置の制約を受ける。ダイアグラムにより、親子関係が容易に把握できる。

2) ウィンドウ間の遷移ダイアグラム作成機能

ユーザの操作と、それに伴うウィンドウ間の遷移をネットワーク形式のダイアグラムで表示する機能。設計ビューでウィンドウ間の遷移を一覧することはできないので、この機能により、複雑なウィンドウ間遷移を避ける。

3) クライアント画面フローダイアグラム作成機能

アプリケーションによっては、1つのウィンドウ内でクライアント画面が次々と入れ替

わっていくものがある。その場合のクライアント画面のフローをダイアグラムで表示する機能であり、ウィンドウ毎にダイアグラムが作成される。

ウィンドウシステムにおいては、ホスト接続型のシングルウィンドウ画面と比較して、画面の構成やウィンドウ間のつながりの把握が困難である。今後、さらにUI評価に必要な検証ビューを考案するとともに、問題点の自動的な指摘や、設計結果の修正のサポートなどを行いたい。

3.5 プロトコル解析との融合^{3) 6)}

U-faceⅡの持つ検証ビューは、あらかじめ定めた観点からのUI評価をサポートするものであり、それ以外の問題点を発見的に見つけることはできない。このような、設計者の予想外の問題点を発見する方法に、ユーザに操作シミュレーションを行ってもらい、その際の動作、発話を記録、分析するプロトコル解析がある。筆者らはU-faceにおいて、操作シミュレーション機能を用いたプロトコル解析を試みたが、実機とシミュレーションとの相違に起因する問題点がいくつか挙がっている³⁾。U-faceは、あくまでUIをシミュレーションするだけでアプリケーション機能と合成して実行する機能がなかったが、U-faceⅡにおいてはそれが可能であるため、今後アプリケーション機能を取り込みながらのプロトコル解析の実験を行っていきたい。

4. おわりに

OS/2のPM上のUI開発ツールU-faceⅡについて述べた。U-faceⅡは、ホスト接続型端末のUI開発ツールU-faceの基本的な考え方を継承しているが、システム設計には大幅な変更を行い、マルチウィンドウアプ

リケーションへの対応、ソースコード出力機能などを強化している。今後、アプリケーション作成におけるU-faceⅡの利用を通じ、高品質なUIの効率的な設計のための方法論を確立していく予定である。

参考文献

- 1) 神場，橋本：マルチビューモデルに基づくユーザインタフェース設計システム U-face，情処研報，Vol.90，No.71（1990）。
- 2) 神場，井関：汎用端末のUI開発ツール U-face(1)－操作規則の階層化－，情処42全大(1991)。
- 3) 伊藤，神場，旭，井関：汎用端末のUI開発ツール U-face(2)－UI評価の観点から，情処42全大(1991)。
- 4) OS/2プレゼンテーションマネージャ，富士ソフトウェア教育出版部著(1989)。
- 5) Marc Adler: A Survey of Windows and Presentation Manager Prototyping Tools, Microsoft Systems Journal, Vol.5, No.1 (1990)。
- 6) H. R. Hartson and E. C. Smith: Rapid prototyping in human-computer interface development, Interacting with Computers, Vol.3, No.1, pp.51-91 (1991)。