

# 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”

神場 知成, 橋本 治

日本電気(株) C&Cシステム研究所

ユーザインタフェース(以下UI)の新設計法としてマルチビューモデルを提案し, それに基づくUI設計システム“U-face”について述べる. マルチビューモデルは, ユーザがシステムに接するときの複数の視点を反映し, UIの設計/検証を複数の側面から行なう方法で, 側面と記述法の区別, 設計ビューと検証ビュー, ビュー間の自動変換などを特徴とする. U-faceは, ホストコンピュータに接続された汎用端末など, メニュー操作を中心とするUIの設計/シミュレーション/検証ツールであり, 画面レイアウト, 画面フロー, 操作手順, 操作モード遷移, キー機能一覧などのビューを持つ.

A User Interface Design System  
based on Multi-View-Model: U-face

Tomonari KANBA, Osamu HASHIMOTO

C&C Systems Research Laboratories, NEC Corporation

This paper proposes Multi-View-Model as a new methodology of UI (User-Interface) design, and describes a UI design system “U-face” which is based on the methodology. Since users look at a system from a lot of points of view, Multi-View-Model integrates informations of multiple aspects of UI. The features of this method are the distinction between aspects and representations, design views and check views, and the automatic transformation functions among some views. “U-face” is a UI design / simulation tool for applications with menu-based UI, such as general purpose terminal screens connected to a host-computer, and has the views of screen layouts, screen transitions, operation sequences, key-bindings etc.

## 1. はじめに

情報処理システムの高機能化、大衆への普及などに伴い、ユーザインタフェース（以下UI）の重要性が急速に高まっている。しかし、UIはシステムと人間との接点に関わるという問題の性質上、定式化が非常に困難であり、さまざまなUI記述法、およびそれらに基づくUI設計法が研究されてきたにもかかわらず、UI設計の一般的な方法論と呼べるものは未だ確立していない。

UI研究の困難さは、分析的な研究が困難であることに起因している。つまり、工学における他の多くの分野のような、問題を分割してそれらを順次解決していく、という手法がとれず、様々な側面のバランスが重要であるということや、特定の側面から見た時の改善が、別の側面から見れば改悪になっているということが頻繁に生じる。

以上のような観点でUI設計に関する従来の研究を見ると、ユーザの側が、全体構成、画面の見易さ、特定の機能を果たすための操作手順など様々な側面から統合的にシステムを見ているのに対し、設計者の側はUIの特定の側面を切りだして個別に検討していることが多かった。例えば操作手順に関しては、状態遷移ダイアグラム、BNF記法、状態遷移表、ベトリネットなどが検討されている<sup>6)7)</sup>。また、UIMS (User Interface Management System)等のUI設計の手法の多くは、UIのappearance(外見)とbehavior(振舞い)という2つの側面だけを区別していた。INTERA/Pでは、ファクシミリ、プリンタなどの制御パネルを直接描画し、その操作と応答をダイアグラムを用いて設計する<sup>8)</sup>。HyperCardでも視覚的デザインと言語による記述を併用する。このように、UIの個々の側面の記述法や設計法は豊富に開発されてきたが、それらがUI全体に与える影響、各側面間のトレードオ

フを考慮しながら統合的に設計を行なう手法については十分に検討が行なわれていない。

上記の観点に基づき筆者らは、UIの複数の側面を統合的に見ながら設計を行なう枠組みである新しい設計法（マルチビューモデル）を検討し、それに基づき汎用端末のUI設計システムU-faceを開発した<sup>1)</sup>。本稿では、まずマルチビューモデルの特徴を述べ、次に、ホストコンピュータ接続型汎用端末画面のUI設計/操作シミュレーション/検証ツールU-faceにおいて、マルチビューモデルの考え方をどのようにして具体化しているかを示す。最後に、U-faceで設定した各ビューの利用法を検討する。

## 2. マルチビューモデル

### 2.1 概要

情報処理システムのUIとは、それらのシステムに人間が接する時にどのように見えるか、という「見え方」と言うことができる。ただし、この場合の「見え方」とは、視覚的なものだけでなく、入力装置のように触覚的なものや、音の出力など聴覚的なものも含んでいる。

ところで、システムの「見え方」は、人間が意識的ないし無意識的に選択した「見方」によって決まるものであるから、UIには人間の特性が直接反映される。例えば、アプリケーションソフトのユーザは次のような性質を持つと考えられる。

- 画面デザイン、操作手順、応答時間、エラーメッセージのわかりやすさなど様々な要素を総合してUIを捉える。
- 上記の捉え方は、時間につれて変動する。これには、ユーザの気分や環境による短時間の変動や、慣れや記憶による長期的な変動がある。
- 上記の捉え方やその変動には個人差がある。

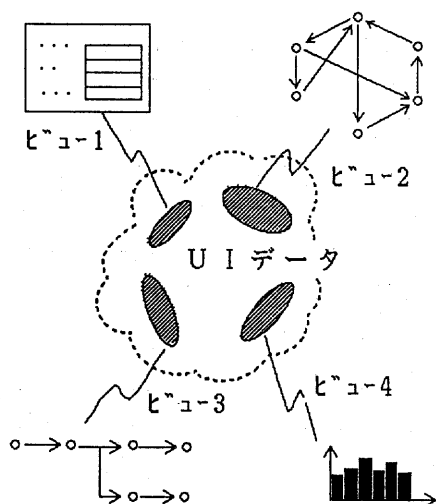


図1. UIのマルチビューモデルの概念図

筆者らは、これらの特徴をUIの「多面性」と考えている。UIは人間の特性に起因する多くの側面を持ち、側面すべてを明らかにすることは、人間の側面すべてを明らかにすること同様、不可能である。

マルチビューモデルはUIが多面性を持つことに着目し、部分の設計が全体に与える影響を色々な側面から把握しながらUI設計を行なう枠組みである。複数の側面から切り出したUIデータをウィンドウシステム上に表示し、側面間の変換や表示方法の変換を利用しながら編集を行なう。(図1参照)

なお、マルチビューという考え方はCADにおける3次元図形表示や、プログラミング環境におけるデータ構造の表示においても存在するが<sup>5)10)</sup>、UIのマルチビューモデルは、これらと次の点で大きく異なっている。

- 従来のマルチビュー表示においては、データ形式は1つであり、その表示方法を適宜変化させて設計の効率化等を図る。
- これに対し、UIは本質的に1つのデータ形式で全体像が記述できず、側面ごとに異なる

ったデータ形式を用いる必要がある。

## 2.2 具体的な特徴

マルチビューモデルの枠組みを次の5点に整理した。

- (1) UIの側面と記述法を区別し、側面と記述法の両方を決めることによって設定されるUIの表示をビューと呼ぶ。
- (2) 設計ビューと検証ビューを分ける。
- (3) ビュー間の自動変換機能を持たせる。
- (4) ビューに階層性を設定する。
- (5) ビューの編集、表示にはウィンドウシステムを利用し、1つのビューに1つのウィンドウを対応させる。

次にそれぞれの項目について説明する。

(1) 従来、UIに対してさまざまな記述法が用いられていたが、それらの中には、着目しているUIの側面が異なるものと、側面は同じで記述法が異なるものが混在していた。例えば、画面の構成要素を図形の種類や色で分類して記述する場合と、タイトルやメッセージといった機能で分類して記述する場合とでは側面が異なる。これに対し、同じ側面に着目していても、例えばユーザの操作手順を、表形式で記述する場合とダイアグラムで記述する場合とでは、記述法が異なる。

マルチビューモデルでは、従来混同されていたUI記述の側面と記述法との区別を明確にする。UIデータは、特定の側面で切り出した後、記述法を決めることにより表示が決まる。この表示をビューと呼ぶ。

(2) 設計ビューと検証ビューとを区別することにより、複数のビューで編集したデータ間に矛盾が出たり、設計者がビューの選択に混乱を生じることを避ける。一般に検証ビューは、設計されたデータを抽象化して全体的に把握する場合に用いるので、視覚的に把握しやすいダイアグラムやグラフ表示を用いる場合が多い。

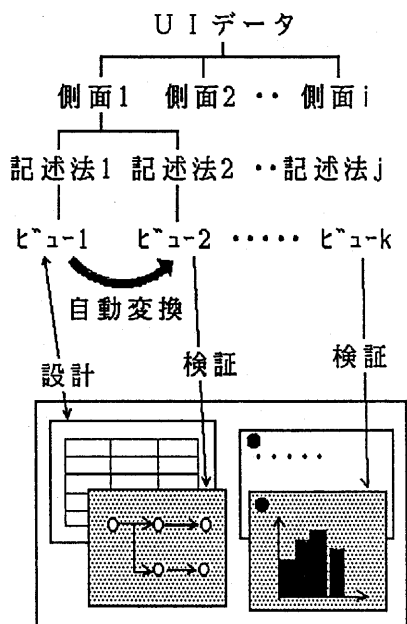


図2 UIデータ，側面，記述法，ビュー

(3) マルチビューモデルではそれぞれの側面がデータ形式を固有に持つため、ビュー相互間の自動変換機能が不可欠である。ビュー間の変換には、一般に次のような特徴がある。

- 同一側面内の記述法の変換は、意味的な変換を伴わないため、容易である。
- 異なる側面間の変換は、不可能または対象領域に依存した知識を必要とする。

(4) ビューは階層性を持つ。階層には、側面の階層と記述の階層とがあり、側面の階層はUIが本来持つものであるから、UI設計に先立ち固定的に設定しておくことができるが、記述の階層は設計の途中でユーザーが自由に設定できることが望ましい。

(5) 以上の考察に基づき、マルチビューモデルの編集、表示にはウィンドウシステムを用い、1つのビューに1つのウィンドウを対応させる(図2参照)。UIの側面、記述法や、設計ビューと検証ビューとの違いによって、ウィンドウのマーキング、色などを区別

する。

### 3. UI設計システムU-face<sup>1)</sup>

マルチビューモデルに基づき、UI設計システムU-faceを開発した。U-faceは、ホストコンピュータに接続された汎用端末の画面UI(ユーザーの操作はキーボードのみ)の設計/操作シミュレーション/検証機能を提供するもので、設計の初期段階において操作イメージを明確に把握、検証することによりUI品質の向上、後戻り工数の低減を目的としている。ここでは、汎用端末画面のUIを具体例として、マルチビューモデルの特徴である、UIの側面と記述法、設計ビューと検証ビュー、ビュー間の自動変換、階層等に関する具体的な検討結果、およびそれらのU-faceにおける実現例を示す。

なお、設計対象をホスト接続型のアプリケーションに設定した理由は、現在利用されているOAシステムの多くがホスト接続型の業務アプリケーションであること、それらのユーザーは一般にコンピュータ等に関する知識はそれほど豊富ではないためUIが問題となりやすいこと、などである。最近ではウィンドウ、アイコンをマウスによって操作するダイレクトマニピュレーション型のUIが主流であるが、それぞれのウィンドウの内部では上記のようなアプリケーション操作が行なわれるため、その応用範囲は広い。

#### 3.1 汎用端末UIの分析

ホスト接続型汎用端末における、業務アプリケーションの画面UIを検討した結果、次のような特徴が明らかになった。

- 一般に、汎用端末のUIはx y軸(平面軸)とt軸(時間軸)で記述できるが、平面軸としては、固定した1画面で切るのではなく、カーソル移動や一部の表示変化などを含む一連の画面群をひと組として表現し、それらの

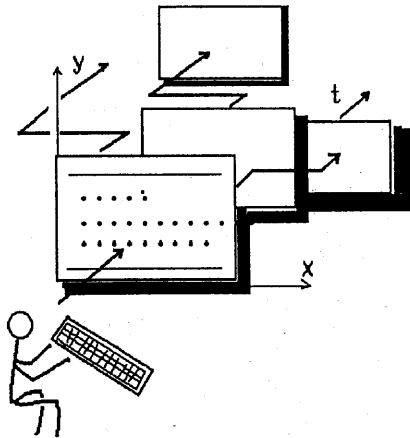


図3. UIのxy軸とt軸

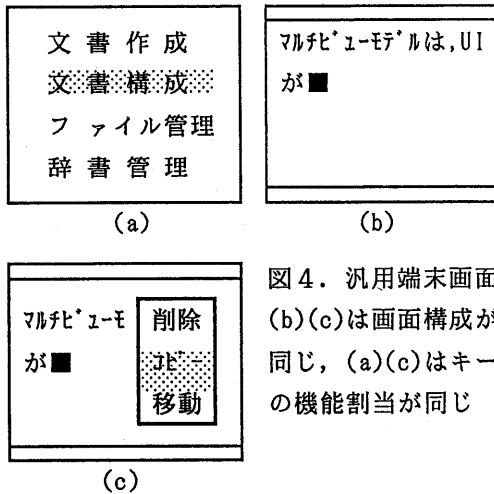


図4. 汎用端末画面  
(b)(c)は画面構成が  
同じ, (a)(c)はキー  
の機能割当が同じ

画面群の間の遷移を時間軸で表現した方がよい(図3参照)。混乱を避けるために、画面内部の操作手順や表示の変化までもを含めた広義の画面を“画面”と呼び、それらの間の遷移を“画面”遷移と呼ぶ。また、“画面”内でカーソル移動やメニューの表示/消去を行なう場合などの操作を操作手順と呼ぶ。

- “画面”遷移には、ツリー構造が内在する場合が多い。
- “画面”内では、カーソル移動やファンクションキー入力などでメニューを選択したり、計算、データベース検索を行ったりする。

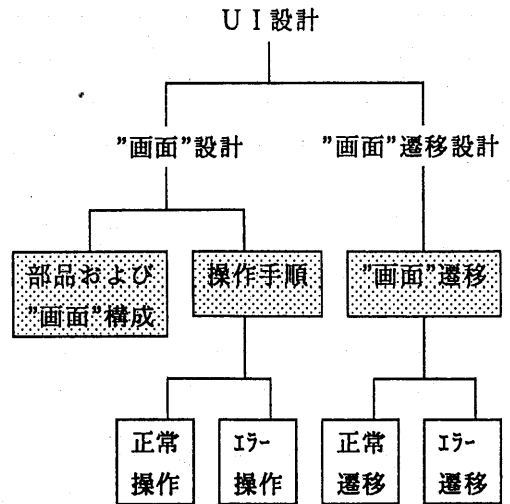


図5. 汎用端末のUI設計の枠組み

- システムの応答は、ユーザの正常操作に対する応答と、エラー操作に対するものに分けることができる。
- 画面の変化を中心とする側面とは別に、キー機能の遷移を示す側面が存在する。これは、システムの各状態でキーに対して割り当てられた機能を示すもので、操作モードの遷移と言い替えることもできる。図4に示した汎用端末の画面例において、画面構成は(b)と(c)とで同じであるが、キー機能の割当は(a)と(c)とで同じである。

### 3.2 UI設計方式

上述の検討に基づき、U-f a c eにおける汎用端末画面のUI設計の枠組みを図5に示すように設定した。長方形で囲んだものがUI設計に関する側面である。“画面”設計に関するものとして、部品および“画面”構成の側面と、操作手順の側面とがあり、操作手順の側面はさらに正常操作、エラー操作という2つの側面を下位に持つ。“画面”遷移設計に関するものとして、“画面”遷移の側面があり、操作手順と同様、正常遷移、エラー遷移の側面に分けることができる。

画面レイアウト設計	
フレーム	メニュー領域 画面登録
業務選択 ← 7000	
1. 文書作成	
2. メール送信 ← 7001	
3. 処理終了	
選択してRETを押して下さい。	

図6. 画面レイアウト設計ビュー

操作手順設計		
画面名:scr1	フィル	正常 エラー 全部
応答覧入力		
カーソル位置	入力	応答
menu1	↓	メニューカーソルを次に移動 [NextMenuItem]
menu1	↑	メニューカーソルを前に移動 [PrevMenuItem]

図7. 操作手順設計ビュー

- 次に、この中で基本的な3つの側面である  
 (1)部品および“画面”構成、(2)操作手順、  
 (3)“画面”遷移の設計について説明する。

(1) 部品および“画面”構成の設計

部品および“画面”構成は、「画面レイアウト設計」のビューで描画ツールを用いて設計する(図6参照)。画面の構成部品は次の3つに分けている。

- フレーム：固定的に表示されるだけで、ユーザによる編集ができない文字列、図形。
- メニュー：ユーザがカーソル移動や文字入力により処理を指示できる領域で、同一“画面”内でも表示されたり消去されたりする。
- ユーザ領域：ユーザがテキスト等を入力、編集する領域。

“画面”構成は、設計したフレーム、メニ

表1. キーワードと機能

キーワード	機能	関数名
メニュー	メニュー表示	DispMenu<>
	メニュー消去	DelMenu<>
メニューカーソル	メニューカーソルを次に移動	NextMenuItem
	メニューカーソルを前に移動	PrevMenuItem
メッセージ	指定領域にメッセージ表示	DispMessage<>
	ビープ音発生	Beep
文書処理	領域選択開始	StartArea
	領域選択終了	EndArea
	指定領域削除バッファに移動	Cut
	バッファ内容を挿入	Paste

ュー、ユーザ領域の名称を組み合わせで登録する。

(2) 操作手順設計

“画面”内の操作手順の設計は、「操作手順設計」のビューで表を用いて行なう(図7参照)。表の項目はカーソル位置、入力、応答の3つから成る。カーソル位置は、入力が行なわれるときのカーソル位置をメニュー項目名で指定するもので、入力はキーの名称で指定する。応答の指定は、記述の容易さ、記述力の高さを同時に満たすものとして、キーワードを利用した選択方式を採用した。これは、U-f a c eが操作シミュレーションのために内蔵する基本機能をキーワード選択によって表示し、その中から適切なものをユーザが選択する方法である。選択したものは、機能の日本語表示と、その機能に対応する関数名との両方で表示される。

U-f a c eで設定したキーワードと、それに対応する機能、関数名を表1に示す(一部省略)。関数名に<>が入っているものは、引数の入力が必要であり、例えばメニュー表

画面フロー設計			
フィルタ <input type="checkbox"/> 正常 <input type="checkbox"/> エラー <input type="checkbox"/> 全部			
ダイアグラム			
前画面	カーソル位置	入力	後画面
scr1	menu1	RET	scr2
scr1		PF1	scr3

図8. 画面フロー設計ビュー

示では DispMenu<menu1> のように補う。

なお、前述のように、操作手順の側面は、さらに下位の側面である正常操作とエラー操作とから構成される。これを実現するために、フィルターという考え方をういた。フィルターを切り換えるためのメニューが、ビュー内の右上の方に「正常」、「エラー」、「全部」の3者択一で表示されている。設計者が「正常」を選んだときには、ユーザの正常な操作に対応するシステムの動作を記述し、「エラー」を選んだときには、ユーザのエラー操作に対応するシステムの動作を記述する。これにより設計全体の見通しを良くすることができる。「全部」を選んだときには、正常操作とエラー操作との両者を合成して表示する。この場合は編集はできない。

### (3) “画面”遷移の操作手順設計

UIの第3の側面は“画面”遷移の操作手順であり、これは「画面フロー設計」のビューで設計する(図8参照)。設計の表は前画面、カーソル位置、入力、後画面の4つの項目から成る。前画面と後画面は、遷移の前後の“画面”名を指定する。カーソル位置と入力の欄の記述法、フィルターの利用法は操作手順設計の場合と同じである。

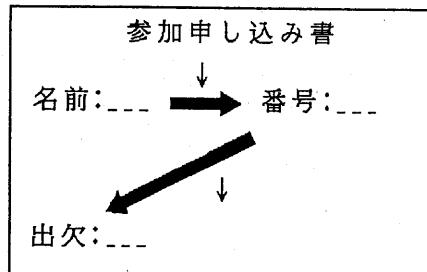


図9. 画面内操作ダイアグラム表示

### 3.3 操作シミュレーション

作成したデータは、操作シミュレーションのビューで検証できる。シミュレーションビューでは、ユーザがキー入力を行なうと、画面フローおよび操作手順の表を参照して画面表示の変更を行なうとともに、キーに対する機能の割り付けを変更する。これにより、実際のシステムを操作する場合のUIを、設計者だけでなく、システムが想定するユーザによっても確認できる。

### 3.4 UI検証方式

一般に、大規模なUIは操作シミュレーションだけでは全体像を把握できないことが多い。U-faceでは、作成したデータに記述法の変換、側面の変換などを行なって、複数のビューで検証できる。それらのビューを検証ビューと呼ぶ。ここではUI検証ビューを、記述法を変換によるものと、側面の変換によるものとに分けて説明する。

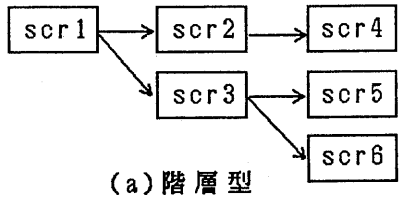
#### (1) 記述法の変換によるUI検証ビュー

##### (a) “画面”検証のビュー

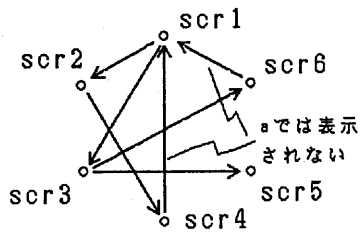
- 「画面内操作ダイアグラム」のビューは、メニュー項目間のカーソル移動等をダイアグラムで視覚的に表示する(図9参照)。

##### (b) “画面”遷移検証のビュー

- 「階層型画面フローダイアグラム」のビューは、画面遷移に内在するツリー構造を取り出して、ダイアグラムで表示する(次ページ図10(a)参照)。



(a)階層型



(b)円型

図10. 画面フローダイアグラム

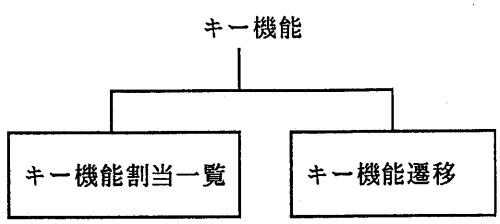


図11. キー機能の側面

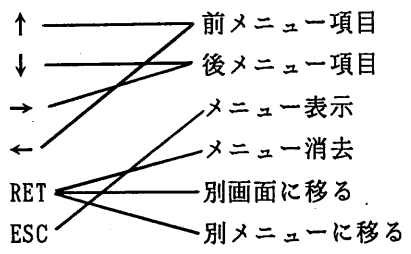
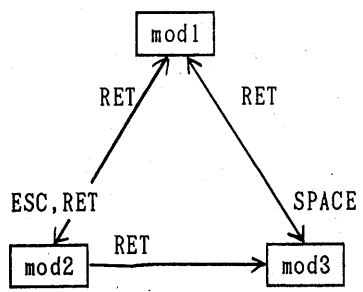


図12. キー機能割当一覧グラフ

• 「円型画面フローダイアグラム」のビューは、画面遷移を円型のダイアグラムで表示する(図10(b)参照)。階層性は表示に現れないが、すべての遷移が明示されるので、階層型ダイアグラムと相補的な性質を持つ。



(a)キー機能遷移ダイアグラム

キー機能割当名	キー	機能
mod1	英数	入力文字を挿入
mod2	↑	メニューカーソルを前項目に移動
	↓	メニューカーソルを次項目に移動
	RET	選択項目を受理
mod3	英数	入力を仮名で挿入

(b)キー機能割当表

図13. キー機能遷移

(2) 側面の変換によるUI検証ビュー

“画面”および“画面”遷移の側面で設計したUIデータは、キー機能の側面に変換することができる。キー機能割当一覧とキー機能遷移の側面とがある(図11参照)。

(a) キー機能割当一覧の側面

操作手順全体を通じてそれぞれのキーに対して割り当てられている機能の一覧を示す側面で(図12参照)、操作手順と“画面”遷移との両者を参照して作成され、「キー機能割当一覧グラフ」のビューに表示する。

(b) キー機能遷移の側面

キーの機能割当の遷移を示すもの(図13参照)で、操作手順と“画面”遷移とを参照して作成し、「キー機能遷移」のビューに表示する。ダイアグラムおよび、その各ノードの内容を示す表から成る。遷移は階層性を持たないので、円型ダイアグラムを用いる。



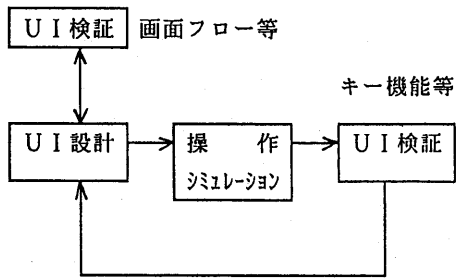


図14. UI 設計の流れ

#### 4. まとめ

U-f a c eのすべてのビューを次に示す。

##### (1)設計ビュー

- 画面レイアウト設計：画面の外観を設計。
- 画面フロー設計（表形式）：画面遷移の操作方法を設計。
- 操作手順設計（表形式）：各画面内でのメニュー表示、消去等の操作方法を設計。

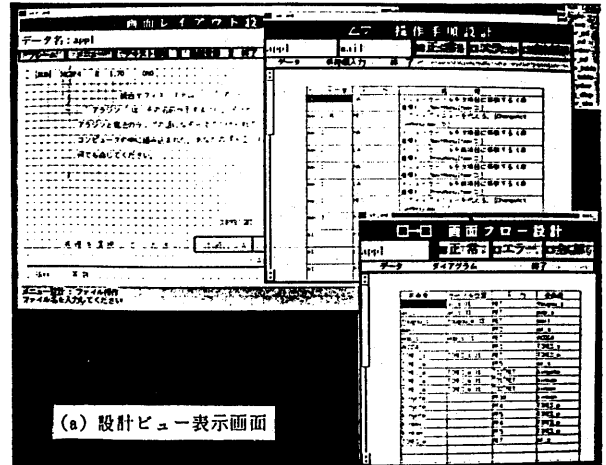
##### (3)操作シミュレーション

実際の操作方法をシミュレーション

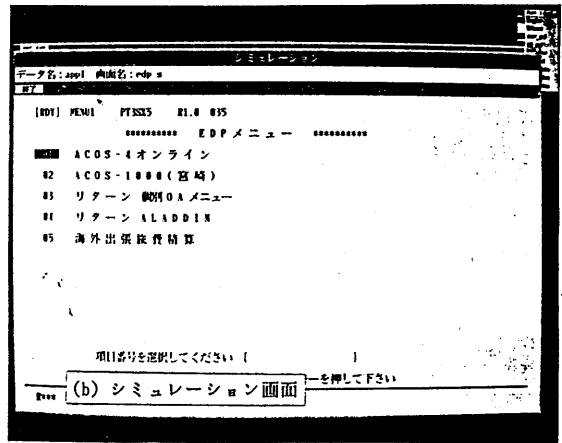
##### (3)検証ビュー

- 画面フローダイアグラム（階層型）：画面フローに内在するツリー構造を検証。
- 画面フローダイアグラム（円型）：画面フローすべてを検証。
- 画面内操作ダイアグラム：各画面内でのメニュー表示、消去等の操作方法を検証。
- キー機能割当一覧グラフ：キーと機能との割当の関係の一貫性を検証。統一性のないキー割当等を発見。
- キー機能遷移ダイアグラム（円型）：操作モードの簡潔性を検証。

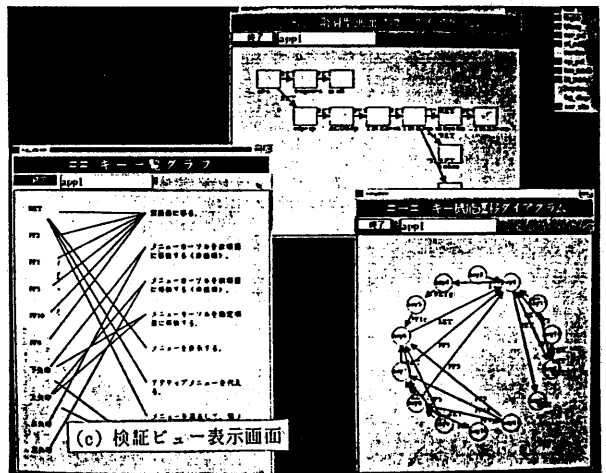
UI 設計者は、検証ビューを参照しながら、設計ビューで設計を行なう（図14参照）。設計、検証ビューのいずれも複数の側面が提供されており、検証ビューによる検証結果は、設計ビューにフィードバックすることにより、繰り返し設計を行なう。



(a) 設計ビュー表示画面



(b) シミュレーション画面



(c) 検証ビュー表示画面

図15. U-f a c e表示画面例

U-f a c eは、NEC製ワークステーションEWS4800/50上にX-Window システム(X11R3)およびUI開発基盤「鼎(V1R2)」<sup>4)</sup>を用いて実現している。設計ビュー(3種)、シミュレーションビュー、検証ビュー(全5種類のうち3種)を表示した画面をそれぞれ前ページの図15に示す。側面と記述法、設計ビューと検証ビューは以下の規則で区別している。

- (1) 側面はタイトル左のマークで区別する。
- (2) 設計ビューと検証ビューは、ウィンドウ背景の色で区別する。検証ビューでは、グレーがかった表示にすることにより、編集不可能であることを示している。

## 5. おわりに

UIの多面性に着目したUI設計法としてマルチビューモデルを提案し、その実現例として汎用端末のUI設計システムU-f a c eを試作した。マルチビューモデルでは、システム設計の初期段階で操作シミュレーション、検証を行いながらUI設計を進めることにより、UI品質の向上、UI変更の後戻り工数の削減による開発効率化が期待できる。今後、開発現場での実際的な利用を通じて、有効性の検証を行ないたい。

また、本稿とは異なるダイレクトマニピュレーション型のUIなどにマルチビューモデルを適用する場合は、新たに側面や記述法の検討を行なう必要があるため、今後、マルチビューモデルの考え方を他の種類のUIに適用して、その有効性を検討する予定である。

## 謝辞

マルチビューモデルのアイデアに関する示唆を与えて下さった、イリノイ工科大学の佐藤啓一先生に感謝致します。

## 参 考 文 献

- 1) 神場, 橋本: 画面シミュレータU-f a c eの試作, 情処全大39, pp.1457-1458(1989).
- 2) 佐藤: ユーザ・インターフェイス設計におけるシステム記述形式, デザイン学研究, No. 76, pp.33-40(1989).
- 3) 谷他: メタユーザインタフェースを有するユーザインタフェース構築支援システム, 情処学論, Vol. 30, No. 9, pp.1200-1210(1989).
- 4) 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, Vol. 31, No. 5, pp.602-611(1990).
- 5) 渡辺他: 状態管理によるマルチビュー表示制御, 情処学論, Vol. 31, No. 1, pp.115-123(1990).
- 6) M.U.Farooq and W.D.Dominick: A Survey of Formal Tools and Models for Developing User Interfaces, Int.J.Man-Machine Studies, Vol.29, pp.479-496(1988).
- 7) H.R.Hartson and Deborah Hix: Human-Computer Interface Development: Concepts and Systems for Its Management, ACM Computing Surveys, Vol.21, pp.5-92(1989).
- 8) O.Hashimoto and H.Miyai: INTERA/P: A User Interface Prototyping Tool, People and Computers III, Cambridge University Press, pp.229-244(1987).
- 9) R.D.Hill: Supporting Concurrency, Communication, and Synchronization in Human-Computer Interaction -The Sassafras UIMS, ACM Trans.on Graphics, Vol.5, No.3, pp.179-210(1986).
- 10) S.P.Reiss: PECAN: Program Development Systemsthat Support Multiple Views, IEEE Trans. on Soft-ware Engineering, Vol.11, No.3, pp.276-285(1985).