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ツール U-face

神場知成† 橋本 治†

ユーザインタフェース (UI) の多面性，多様性を考慮した設計モデルとして，マルチビューモデルを提案する．それに基づいてホストコンピュータ接続型汎用端末の UI 設計ツール U-face を試作した．従来開発された UI 設計ツールの多くは，設計効率化を主目的としており，ツールを用いて作成される UI の品質向上をサポートする機能が十分提供されていない．しかし，大規模な UI になると，プロトタイプの試用だけで問題点を見つけることはかなり困難である．マルチビューモデルは，UI をいろいろな側面から検証してバランスを考えながら設計するための設計モデルである．特徴として，(1)UI の側面の分類と，それぞれの側面に対する複数の記述法の提供（これによりビューが決まる），(2)設計ビュー（設計用）から検証ビュー（評価用）への自動変換機能，(3)ウィンドウシステムの利用による側面間のバランスの検討，の3つがある．これに基づいて試作した U-face は，汎用端末の UI 設計／操作シミュレーション／検証ツールである．汎用端末の分析結果に基づき，画面レイアウト，各画面内での操作手順，画面フローという3つの側面に対応する設計ビューと，操作シミュレーション，操作手順ダイアグラム，画面フローダイアグラム，機能割当一覧グラフ，モード遷移ダイアグラムなどの検証ビューを設定した．代表的なアプリケーションの UI を U-face 上で試験的に実現し，「従来の手法で早期発見困難な問題が設計時に見つけやすくなる」，「UI の側面間のバランスをビジュアルに比較できる」などの効果を確認した．

U-face: a User Interface Design Tool Based on Multiview Model

Tomonari Kanba† and Osamu Hashimoto†

A novel user interface (UI) design model called a multiview model is proposed, and a UI design tool, U-face, developed with this model is shown to contribute to improved UI quality. The multiview model is a design model to design well-balanced UI by checking it from various aspects. It is based on the following principles. (1) Various UI aspects are described in a variety of ways, and each description is called a view. (2) Design views are automatically transferred to verification views. (3) Each view is displayed on a window of a multi-window system. U-face, developed with this model, is a UI design tool for generic terminals and it consists of design views and verification views. The design views consist of a screen layout view and operation rule view. The verification views include an operational simulation view, a screen flow diagram, and a mode sequence diagram, and so on. The efficiency of U-face is verified by the experiment of building a typical application and finding its UI problems.

1. はじめに

情報処理機器の高機能化，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，という手法があまり有効ではない．対象を包括的に捉えた時の様々な側面のバランスが最優先されるという場合が多く，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．

ところで，UI 設計における有効な手法の1つとして広く利用されているものに，UI プロトタイピングがある[1),2)]．これは，ソフトウェア仕様設計の初期段階で UI 部分だけを切り出してシミュレーションを行

† NEC C&C システム研究所
C&C Systems Research Laboratories, NEC Corporation

い，不適当な部分を繰り返し修正するという，繰り返し設計の考えに基づくものである．しかし，前述のような観点で，プロトタイピング機能を持つ従来の UI 設計ツールを見ると，それらは設計の効率化に主眼を置いており，ツールを用いて作成される UI の品質向上をサポートする機能が十分提供されていない．例えば，谷越らは，オブジェクトに対して外形，イベントとアクションを設定することによる UI 構築支援システムを作成し[3]，橋本らはパネル型システムの UI シミュレータを開発している[4]が，いずれにおいても UI のビジュアル設計と操作シミュレーションを主眼にしており，作成した UI プロトタイプは，人間の実際の操作だけで評価することを前提としている．しかし，一般に作成するソフトが大規模になれば UI 部分の操作も複雑であり，その問題点を人間が試用するだけで見つけることは非常に困難である．操作とそれに対するレスポンスの良しあしなどはわかっても，全体を通じての画面フローの構成や，モード遷移の構造などは簡単には把握できないからである．

以上の観点に基づき本論文では，UI の複数の側面を統合的に見ながら設計，操作シミュレーション，および検証を行う枠組みである新しい UI 設計モデル(マルチビューモデル)を提案し，それに基づいて試作した，ホストコンピュータ接続型汎用端末画面の UI 設計ツール U-face について述べる[5)~7)]．U-face は UI の品質向上を主目的としているが，UI の問題点発見を容易にし，UI の各側面間のバランスをビジュアルに確認する機能は，UI 品質の効率的な改善，さらに，UI 変更の後戻り工数の削減にも貢献するはずである．

なお，UI 設計におけるマルチビュー，あるいはそれに類する言葉としては，CAD (Computer Aided Design) におけるマルチビュー表示[8]，Avrahami による Two-View Approach[9]，England による Multiple-View User Interface Design[10] などがある．CAD におけるマルチビューとは，1つの図形を見る方向や拡大倍率を変えて表示するものであり，UI 自身をマルチビューというモデルで捉えようとする本論文のアイデアとは異なる．また，Avrahami や England によるものは，いずれも UI を複数の記述法で設計可能にすることにより設計者の自由度を上げるということを目標としており，本論文で述べる UI 品質向上のための検証ビューなどの考え方とは異なっている．

2. マルチビューモデル

2.1 概　　要

情報処理機器の UI とは，それらの機器に人間が接する時の「見え方」である．そしてその「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，エンドユーザがアプリケーションソフトに接するときの捉え方には，次のような性質があると考えられる．

- 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．

筆者らは，これらの特徴を「UI の多面性，多様性」と考えている．他の多くの分野における設計行為が主として物理的な制約を満たすことを目的としているのに対し，UI においては人間の多面的な性質に適合するような設計をすることが求められている．

このように UI が多面性，多様性を持つことに着目し，UI をいろいろな側面から検証してバランスを考えながら設計するための設計モデルが，マルチビューモデルである．設計者は設計用のビューから UI を設計し，コンピュータがその UI データを検証用のビューに変換してウィンドウシステム上に表示する．設計者はコンピュータが提供する複数のビューを見ながら，バランスのとれた UI を設計する．

2.2 具体的な特徴

マルチビューモデルの特徴として，次の3つがある(**図1**参照)．マルチビューモデルは設計者がコンピュータを利用して作業を行うことを前提とした設計モデルである．

(1) ビュー

コンピュータは，対象とする UI に応じて，複数の側面と，各側面に対する複数の記述法を提供する．側面と記述法との両方を決めることによって実際の表示が決定し，これをビューと呼ぶ．

(2) 設計ビューと検証ビュー

ビューには，設計ビューと検証ビューとがある．コンピュータは，設計ビューから検証ビューへの自動変換機能を提供する．

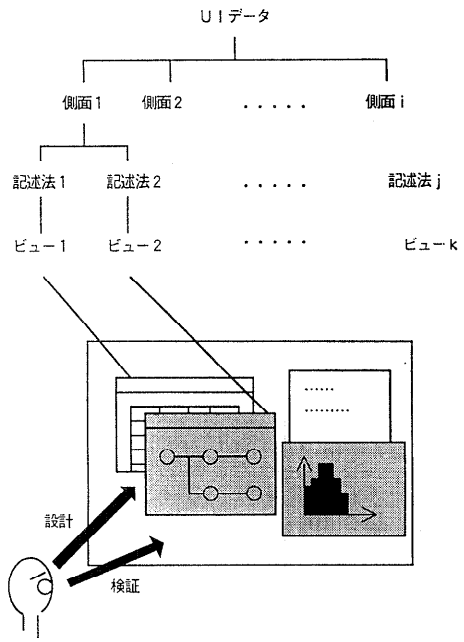


図 1 マルチビューモデル
Fig. 1 Multiview model.

（3） ウィンドウシステムの利用

1つのビューは，ウィンドウシステムにおける1つのウィンドウ上に表示される．

次にそれぞれの項目について説明する．

（1） 従来，さまざまな角度からUIが論じられているが，それらが論じている対象として，問題にしているUIの側面が異なるものと，側面は同じであるが記述法だけが異なるものとが混在していた．例えば画面レイアウトを論じる場合であっても，画面の構成要素を図形や色といった見かけの特徴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意味的な特徴で分類して記述する場合とではUIの側面が異なる．これに対し，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，記述法が異なるだけである．

マルチビューモデルでは，従来混同されていたUIの側面と記述法とを区別する．コンピュータが提供するビューは，側面と記述法の両者によって決定される．具体的にどのような側面と記述法があるかということは，設計対象ごとに分析に基づいてあらかじめ決め，コンピュータに組み込む必要がある．

（2） ビューには，設計ビューと検証ビューとがある．設計ビューとは，設計者がUI設計を行うためのものであり，検証ビューはそれをユーザの立場で把握，検討する際の材料とするためのものである．設計ビューから検証ビューへの設計データの変換機能は，コンピュータが提供する．一般に，検証ビューは設計されたUIデータを分析，抽象化して大局的に把握するためのものであり，記述法としては，視覚的に把握しやすいダイアグラムやグラフ表示が適している．

（3） 1つのビューは，ウィンドウシステムにおける1つのウィンドウ上に表示される．UIにおいては，1つの側面における改良が他の側面における改悪になるというようなことがあり，それらのバランスを取りながら設計を進めることが重要である．ウィンドウシステムの特徴を利用することにより，各側面を同時に把握することができ，設計者が側面間のバランスを考えるのに適している．各ウィンドウは，側面の違い，記述法の違いがわかるようにウィンドウ枠の表示などを区別する．

3. UI 設計ツール U-face[5)~7)]

設計モデルとしてマルチビューモデルを採用し，UI設計ツール U-face を開発した．U-face は，ホストコンピュータ接続型汎用端末の UI 設計，操作シミュレーション，および検証機能を提供するものである．設計対象は次のような条件を満たすものである．

- 画面はシングルウィンドウで構成され，メニュー選択，キー入力などを中心に操作が進む．
- 入力デバイスとしてはキーボードやマウスが利用可能である．マウスは，メニュー選択や項目への入力の際にメニュー項目や入力欄を指示する目的で使用される．

以下では，マルチビューモデルの特徴の，U-faceにおける具体的な実現方法を示す．なお，設計対象をホスト接続型の汎用端末に設定した理由は，現在利用されている OA システムの多くがこの形態であること，それらのユーザは一般にコンピュータ等に関する知識はあまり豊富ではないため UI が問題となりやすいこと，などである．また，マルチウィンドウシステムについても，各ウィンドウ内部の動作を設計するためには，本汎用端末 UI 設計ツールが利用できる．

3.1 汎用端末 UI の分析

社内で利用されている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．

（1） 一般に，汎用端末の画面遷移は xy 軸（平面軸）と t 軸（時間軸）で記述できるが，xy 軸におい

ては，固定した1画面で切るのではなく，カーソル移動やポップアップメニュー表示などを含む一連の画面群をひと組として表現し，それらの画面群の遷移を t 軸で表現したほうが良い（**図2**参照）．以下では，特にことわりのない限り，上述の一連の画面群を一画面として扱う．それらの間の遷移を画面フローと呼ぶ．また，一画面内でカーソル移動やメニューの表示／消去などを行う場合の操作を操作手順と呼ぶ．

（2） 端末からの応答は，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とに分けることができる．エラー操作に対する応答としては，エラーメッセージが出る場合と，単に入力を全く受け付けない場合とがあるが，ユーザインタフェースの観点からはエラーメッセージが出たほうが良い．

（3） 画面フローは，ツリー構造が基本となっている場合が多い．つまり，ハイパーメディア型のUIを持つソフトウェア[11]では画面間がネットワーク型に連結されていて，開始画面や終了画面などの区別がない場合が多いのに対し，汎用端末UIではあらかじめ画面の提示順序が決まっており，それが分岐しているだけの場合が多い．

（4） 画面の変化を示す側面とは別に，モード遷移を示す側面が存在する．モードとは，キー押下やマウスクリックなどの操作に割り当てられている機能の構造である．例えば，**図3**に示した汎用端末の画面例において，(b)と(c)はメニューを表示しているかどうかの違いだけであるので，前記の基準に従えば同一画面として扱うことができるが，(a)と(c)は両方ともメニュー選択のモードになっており，同一モードとして扱うことができる．

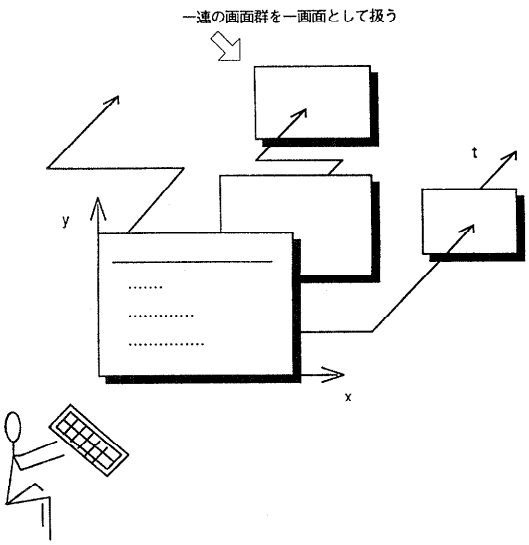


図2 汎用UIの xy 軸と t 軸
Fig. 2 XY axis and T axis of generic terminal UI.

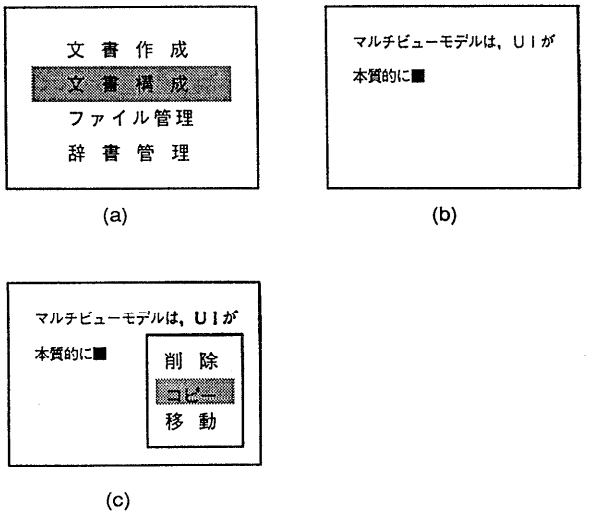


図3 画面構成とモード
((b)と(c)は画面構成が同じ，(a)と(c)はモードが同じ)
Fig. 3 Screen composition and mode.

3.2 UI設計方式

上述の検討に基づき，ビューとして次の3つを設定した（**図4**参照）．3.1節の(1)で述べたことに基づき画面フローの設計と一画面内の操作手順の設計とを分けたこと，(2)で述べたことに基づき画面フローと画面内操作手順の設計を正常操作とエラー操作に分けたことが特徴である．

- 各画面の画面レイアウトビュー
- 各画面内の操作手順表ビュー
- 画面フロー表ビュー

以下でそれぞれについて説明する．

（1） 画面レイアウトビュー

画面レイアウトの側面を，直接表示の記述法で設計するビューである．ビューは，レイアウトを行う土台となる描画領域と，設計用部品メニューとから成る（**図5**参照）．設計用部品は次の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．

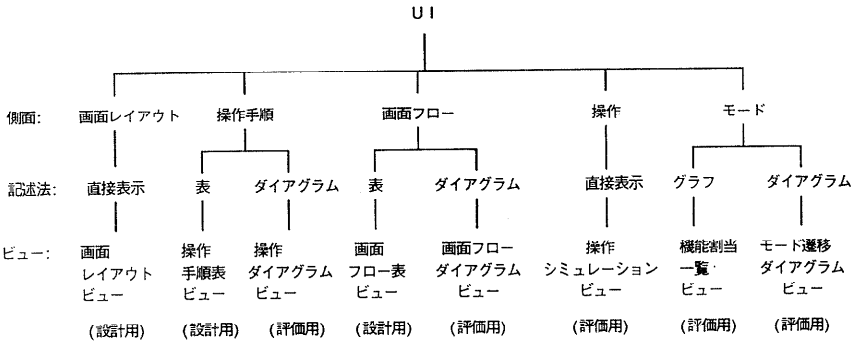


図 4 U-face におけるマルチビュー
Fig. 4 Multiview in U-face.

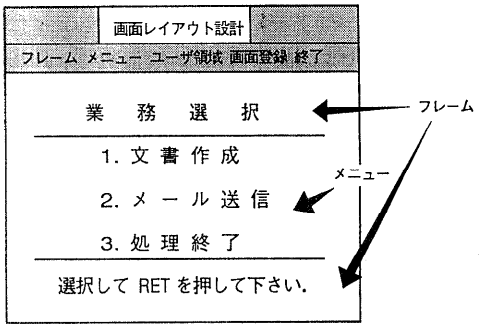


図 5 画面レイアウト設計ビュー
Fig. 5 Screen layout design view.

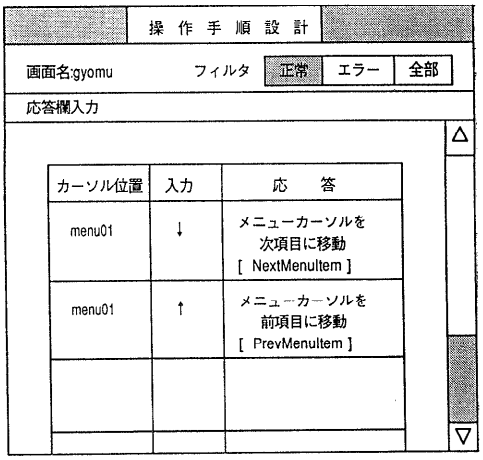


図 6 操作手順設計ビュー
Fig. 6 Operation sequence design view.

● ユーザ領域：ユーザがテキスト等を入力，編集する領域.

画面レイアウトを行う際は，まず部品を配置してからそれぞれの部品に名称をつけて登録し，それらの部品の集合として各画面を登録する.

（2） 操作手順表ビュー

各画面内でのユーザの操作手順を，表形式の記述法で設計するビューである．一画面にひとつの表がある(**図 6** 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，入力が行われる時のカーソル位置に対する制約条件を示すものであり，メニュー名またはメニュー項目名を利用して指定する．例えばこの欄に menu 01 と記述しておき，入力欄に RET と書いてあったとすると，カーソルがメニュー menu 01 の上にあってリターンキーの入力が行われた時に限り，応答が定義できる．入力欄はキーの名称で指定する．例えばアルファベットはそのまま記述すれば良いし，リターンキーなどに関しては，RET などあらかじめ記述法を定めている．マウスによるクリックに対しては，m_click と記述すれば良い（本設計対象におけるマウスの利用法はこれだけである)．応答欄の指定は，キーワード選択による説明文選択方式を用いた．これは，あらかじめ用意されたキーワードを選ぶことによって機能を選択するもので，選択した機能はその説明文とともに表の中に書き込まれる．ここで指定した機能は，後の操作シミュレーションの際にシステムがシミュレートする必要があるが，キーワードを利用して選択した機能に対しては，U-face がシミュレーション機能を保証している．U-face で設定したキーワードと，それに対応する機能，関数名を一部，**表 1** に示す．関数名に<>が入っているものは，引数の入力を必要とする．機能を示す日本語の<>内に引数を入れれば，その結果が実際の関数の引数にも反映する.

なお，操作手順表は正常操作とエラー操作とに分けて記述するようになっている．正常操作とエラー操作

表 1 キーワードと機能
Table 1 Keywords and functions.

キーワード	機能	関数名
メニュー	メニュー<>を表示	DispMenu<>
	メニュー<>を消去	DelMenu<>
メニューカーソル	メニューカーソルを次項目に移動	NextMenuItem
	メニューカーソルを前項目に移動	PrevMenuItem
メッセージ	領域<>にメッセージ<>表示	DispMessage<,>
	ビープ音発生	Beep

とを分け，エラー操作設計の必要性を設計者に意識してもらうようにすることにより，UI 向上が期待できる．この機能を実現するために，フィルタという手法を用いた．フィルタは設計する UI の範囲を切り換える機能で，このためのメニューが，ビュー内の右上のほうに「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が合成して表示される．なお，この場合は編集はできない．

（3） 画面フロー表ビュー

画面フローの側面を表形式の記述法で設計するビューである（**図 7** 参照）．表は前画面，カーソル位置，入力，後画面の 4 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，フィルタの利用法は操作手順設計の場合と同じである．

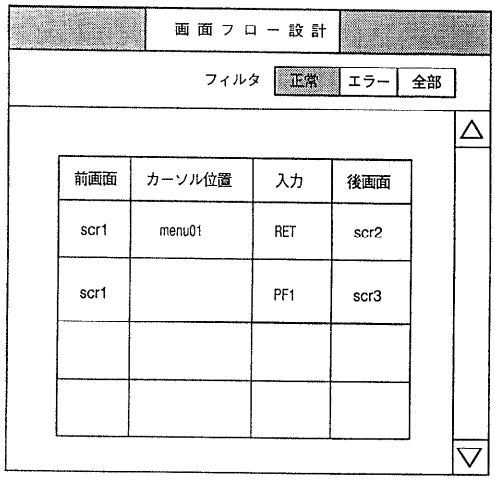


図 7 画面フロー設計ビュー
Fig. 7 Screen flow design view.

3.3 操作シミュレーション

作成した UI は，操作シミュレーションビューで検証することができる．ビューとしては，設計した画面レイアウト，操作手順，画面フローに従って動作する画面が直接表示される．ユーザがキー入力またはマウスクリックを行うと，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うとともに，キーやマウスに対する機能の割り付けを変更する．このビューは，設計者だけでなく，作成しようとするソフトウェアを実際に使用する予定のユーザによっても確認してもらうことが望ましい．

3.4 UI 検証方式

操作シミュレーションによる問題点の発見は UI 向上に必要かつ有効な方法であるが，一般に大規模な UI の品質向上をこの方法だけで行うことは非常に困難である．操作手順数が多くなって確認に時間がかかるという問題だけでなく，全体の構造を大局的に把握することは人間にとってかなり困難だからである．このような場合に，ユーザは漠然とわかりにくさを感じたりするが，「なんとなく使いづらい」としか表現できない．U-face ではマルチビューモデルに従い，作成した UI に記述法の変換，側面の変換などを自動的に行って検証用のビューを提供することでこの問題に対処している．ここでは UI 検証ビューを，記述法の変換によるものと，側面の変換によるものとに分けて説明する（図 4 参照）．3.1 節の（3）で述べたことに基づき画面フローをツリー状に表示する機能を設けたことと，（4）で述べたことに基づきモード遷移の検証ビューを設けたことが特徴である．

（1） 記述法の変換による UI 検証ビュー

●操作ダイアグラムビュー

各画面内での操作手順の側面は，記述法として表を用いて設計するが，それによるカーソルの動きなどを一目で把握することは難しい．このビューは，各画面の操作手順の側面をダイアグラムでの記述に変換してカーソルの動きを容易に把握できるようにしたものである（**図 8**）．カーソルの動きは，ユーザが使用する際の目の動きに対応するので，このダイアグラムの矢印を目で追う際に不自然な動きにならないように設計することが重要である．

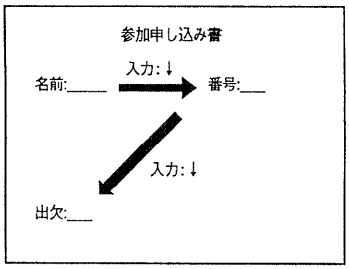


図 8 画面内操作ダイアグラム
Fig. 8 Operation diagram within a screen.

● 画面フローダイアグラム

画面フローの側面も操作手順同様，表で設計するのでその構造を視覚的に把握することができない．3.1節で述べたように汎用端末 UI では画面フローがツリー構造を持っていることが多く，その構造が複雑であると使いにくさの原因になる．このビューは，画面フローの側面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(**図 9**)．なお，ダイアグラムを表示するためには，設計者はダイアグラムツリーのルートとなる開始画面を指定する必要がある．設計者はツリーの分岐のバランスを考慮して設計することが重要である．

（2） 側面の変換による UI 検証ビュー

設計の段階では陽に見えないが，3.1節で述べたように，汎用端末の UI として重要な要素にモードがある．ユーザがシステムを操作するときにモードがどのように変化するかということは，使いやすさ，覚えやすさ等に大きく影響する．U-face は，設計した UI データをモードの側面に自動的に変換する．モードの記述法としては，機能割当一覧をグラフで示すものと，モード遷移をダイアグラムで示すものとがある．

● 機能割当一覧グラフビュー

モード全体を通してのキー割当の一覧を示すビューである．キーに対する機能割当の一貫性を判断する際に有効である．操作手順全体を通じてそれぞれの入力

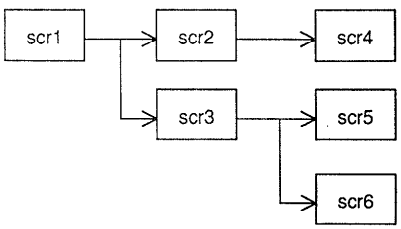


図 9 画面フローダイアグラム
Fig. 9 Screen flow diagram.

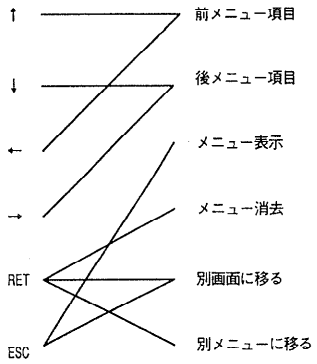
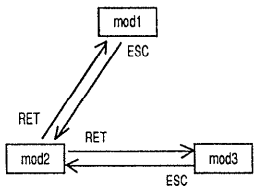


図 10 機能割当一覧グラフ
Fig. 10 Key binding graph.

（キー押下またはマウスクリック）に対して割り当てられている機能の一覧をグラフで示す(**図 10**)．このグラフで1つの入力に割り当てられている機能が複数あり，それらの間に意味的な一貫性がない場合はユーザの混乱の原因となる．1つの機能に複数の入力が割り当てられている場合は入力の柔軟性を示す場合もあるが，混乱を招く場合もある．それらの詳細な判断は設計者に委ねられている．

● モード遷移ダイアグラムビュー

操作の簡潔性を判断する際に有効である．モード遷移をダイアグラムで示し，その各ノードにおける入力と機能との対応関係を付属の表で示す(**図 11**)．モードの数が増えすぎると，ユーザにとって操作を覚えにくくなる．モード遷移は通常階層性を持たないので，ノードを円周上に均等に配置する円型ダイアグラムによって表示している．



モード名	入力	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
mod3	英数	入力を仮名で挿入

図 11 モード遷移ダイアグラム
Fig. 11 Mode sequence diagram.

なお，以上のビューを利用してUIを設計する際には，設計者は各検証ビュー間のバランスを考慮することが必要である．例えば，一般に画面フローの分岐数を減らそうとすると一画面内での操作が複雑となる．1つのモードでのキー割当を単純にしようとするとモードの数が増えてしまう．U-faceはこれらの考慮すべき側面をコンピュータが自動的に生成して設計者に見せるが，それらの情報をもとに全体のバランスを判断するのは設計者である．

4. 試用結果

U-faceは，NEC製ワークステーションEWS 4800/50上にX-Windowシステムおよび UI 開発基盤「鼎」[12]を用いて実現している．この有効性を調べるために，汎用端末の代表的なアプリケーションのUIをU-faceの設計ビュー上で作成し，検証ビューを利用してUI上の問題点を抽出する実験を行った．作成したUIの，U-faceによる分析結果を**表2**に示す．

この結果によると，モード数が14とやや多すぎるようなので，この原因を調べるために，各モードにおけるキー割当を，モード遷移ダイアグラムに付属する表で調べてみた．その結果，似たような場面でキー割当が少しだけ異なっているために，それらが別のモードとしてカウントされ，全体のモード数が増えていることがわかった．例えば，横方向に並んだメニューでカーソルが一番右にある時に，ある画面では右矢印キーを押すと何も動作しないのに対し，ある画面では一番左の項目にカーソルが移動する，というようなことがあった．これは，操作の一貫性という観点からも問題がある．また，あるメニュー画面においてはファンクションキーにより前画面に戻ることが可能であるのに対し，別のメニュー画面では不可能で，これらが別のモードとして分類されていることがわかった．これらを別のモードとして扱うべきかどうかという点に関しては議論がわかれると思うが，UIとして良くないことは明らかである．これらの点を修正することにより，モード数は9に減少し，操作が簡潔になったことがわかる．この場合は他の側面に対する影響はない．従来の設計ツールを用いて設計を行った場合，このような問題点は発見するのが困難であり，製品として実装を行って利用を繰り返すうちに問題になる．その結果，製品を修正するために大きな時間とコストがかかる．U-faceでは，マルチビューモデルの検証ビューを利用することにより，設計時にはわからないモードの問題点などを見つけることを容易にしている．これは，問題点の発見と修正のサイクルを短くし，結果としてUI向上につながる．

表2 汎用端末UIの分析結果
Table 2 Analysis result of generic terminal UI.

画面数	17	
すべての画面に対する操作手順数の合計	35	
1入力に割り当てられた機能数	最大	5 (RETキー)
	最小	1
	平均	1.5
1機能に割り当てられた入力数	最大	6 (別画面への移動)
	最小	1
	平均	2.3
モード数	14	

他の側面とのトレードオフが問題になる例としては，1画面におけるメニュー項目数の問題がある．ここで対象としたアプリケーションにおいて，1画面で選択可能な最大メニュー項目数は4であった．このため，各画面のレイアウトおよび操作手順ダイアグラムはかなり単純になっているが，画面フローを見るとツリーの枝が長く，目的とする1つの処理をするまでに最大8画面遷移する場合がある．これらに関しては，各画面におけるメニュー項目の数をふやすことによる，画面レイアウトおよび操作手順ダイアグラムの複雑化と，画面フローの単純化とのバランスを考慮する必要がある．従来の設計ツールでは，このような場合，各画面のメニュー項目の数と画面フローが関連していることはわかっても，ある画面におけるメニュー項目数を変えたときに操作手順数がどう変わるかということは手作業で調査しなければわかりづらいが，U-faceでは検証ビュー間の比較をビジュアルに行うことができ，作業の容易さの点で大きく優っている．

U-faceにおいて操作手順表，画面フローダイアグラム，機能割当一覧グラフを表示した画面を**図12**に示す．

5. おわりに

UIの多面性に着目したUI設計モデルとしてマルチビューモデルを提案し，それに基づいて試作した汎用端末用UI設計ツールU-faceについて述べた．マルチビューモデルは，UI設計と操作シミュレーションの機能に加え，複数の側面からUIを大局的に把握

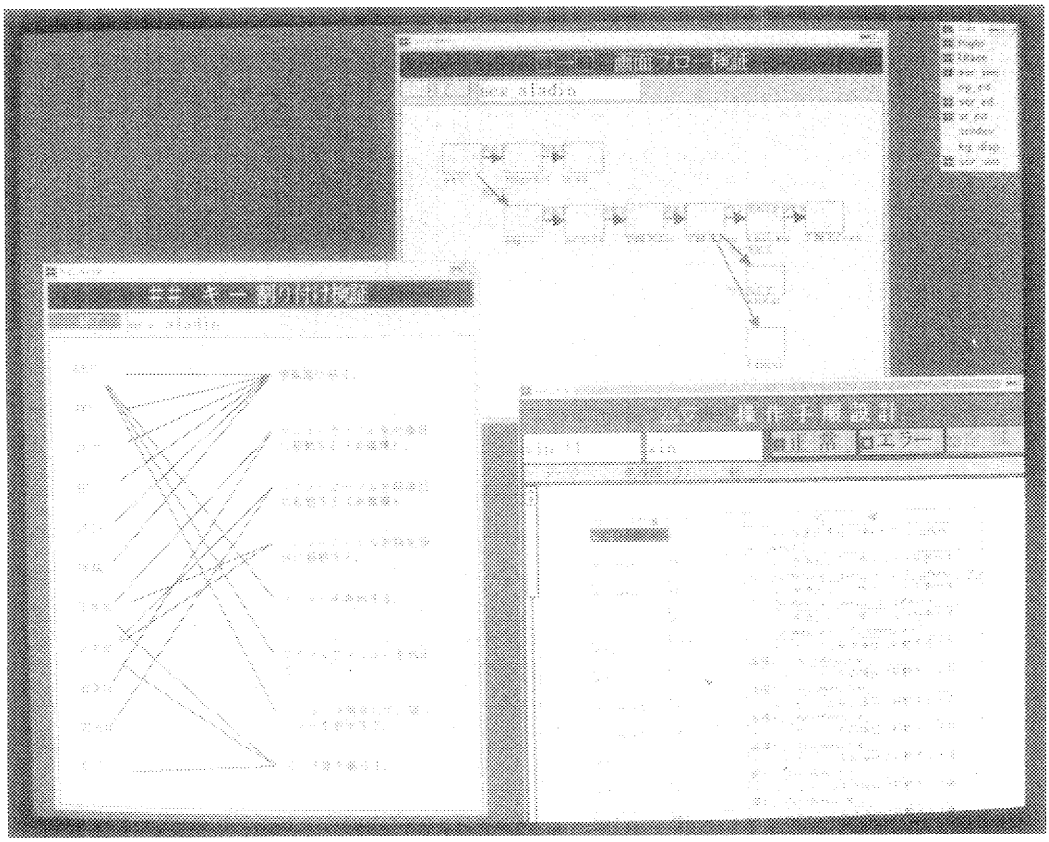


図 12 U-face
Fig. 12 U-face.

するための検証ビューを提供することにより，UI 品質の効率的な改善，さらに，UI 変更の後戻り工数の削減による開発効率化を目的としている．実際に U-face を利用して既存ソフトの改善点を発見する実験を行った．今後，さらに開発現場での実際的な利用を通じて，U-face の有効性検証をさらに進めるとともに，マルチビューモデルに基づく他の UI 設計ツールを開発していく予定である．

謝辞 本研究の機会を与えてくださった C&C システム研究所山本所長，ターミナルシステム研究部西谷部長，川越課長，数多くの有益な助言をくださった関西 C&C 研究所宮井課長，井関課長，旭主任に感謝いたします．また，マルチビューモデルのアイデアに関して議論してくださった，京都工芸繊維大学の佐藤啓一先生に感謝いたします．

参 考 文 献

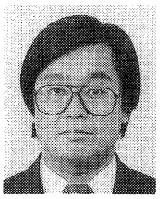
1) Farooq, M. U. and Dominick, W. D.: A Survey of Formal Tools and Models for Developing User Interfaces, *Int. J. Man-Mach. Stud.*, Vol. 29, pp. 479-496 (1988).
2) Hartson, H. R. and Hix, D.: Human-Computer Interface Development: Concepts and Systems for Its Management, *ACM Comput. Surv.*, Vol. 21, pp. 5-92 (1989).
3) 谷，荒井，谷越，谷藤，横山：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，情報処理学会論文誌，Vol. 30, No. 9, pp. 1200-1210 (1989).
4) 橋本，宮井：ユーザインタフェースシミュレータ INTERA，情報処理学会論文誌，Vol. 31, No. 10, pp. 1497-1504 (1990).
5) 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"，情報処理学会ヒューマンインタフェース研究会報告，Vol. 90, No. 71 (1989).
6) U-face: a User Interface Design System Based on Multiview Model, *Proc. HCI Int. '91*, pp. 684-688 (1991).
7) 神場，橋本，井関，宮井：コンピュータによる

ユーザインタフェースデザイン，デザイン学研究，No. 84, pp. 43-48 (1991).
8) 渡辺，福島，富田，後藤：状態管理によるマルチビュー表示制御，情報処理学会論文誌，Vol. 31, No. 1, pp. 115-123 (1990).
9) Avrahami, G., Brooks, K. P. and Brown, M. H.: A Two-View Approach to Constructing User Interfaces, *Comput. Gr.*, Vol. 23, No. 3, *SIGGRAPH'89 Conf. Proc.*, pp. 137-146 (1989).
10) England, D.: MUD: Multiple-View User Interface Design, *INTERACT '90*, pp. 613-618 (1990).
11) HyperCard ユーザーズマニュアル，アップルコンピュータジャパン(株) (1988).
12) 暦本，垂水，菅井，山崎，猪狩，森，杉山，内山，秋口：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).
13) 佐藤：ユーザ・インタフェー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).

(平成4年4月8日受付)
(平成4年11月12日採録)

神場 知成（正会員）

1962年東京都生．1986年東京大学大学院修士課程(電子工学)修了．同年NEC入社．以来，C&Cシステム研究所において，ユーザインタフェース，マルチメディア等の研究に従事．現在，C&Cシステム研究所ターミナルシステム研究部主任．



橋本 治（正会員）

1958年東京都生．1983年早稲田大学大学院修士課程（電気工学）修了．同年NEC入社．以来C&Cシステム研究所にて，ユーザインタフェースの研究に従事．1990年～1991年米国カーネギーメロン大学に客員研究員として留学．1992年度東京工業大学非常勤講師．現在，C&Cシステム研究所ターミナルシステム研究部主任．