

ユーザインタフェース設計ツール

— 画面設計支援 —

福岡 久雄、宮崎 一哉、辻 順一郎、坂下 善彦

三菱電機㈱ 情報電子研究所

高機能ワークステーション上のアプリケーション・プログラムでは、ユーザインタフェースとしてウィンドウ、メニューなどを駆使した視覚型ユーザインタフェースを採用し、ユーザの使い勝手の向上を図ったものが多い。このようなアプリケーションの開発効率を向上させるためには、ウィンドウやメニューなどの設計/変更を容易に行えることが必要である。そこで我々は、ウィンドウやメニューの仕様定義をアプリケーション本体から切り離して行い、それらの表示形態や表示順序の設計/変更をアプリケーション本体とは独立に行えるユーザインタフェース設計ツールを開発した。

本稿では、ユーザインタフェース設計ツールの構成と機能について述べる。

User Interface Design Tool

— Screen Design Aid —

Hisao FUKUOKA, Kazuya MIYAZAKI, Junichiro TSUJI, Yoshihiko SAKASHITA

Information Systems and Electronics Development Laboratory

Mitsubishi Electric Corporation

5-1-1 Ofuna, Kamakura City

247 Japan

Most application programs on current high performance workstations adopt visual user interfaces using windows, menus and so on. Design and modification of these displayed objects should be easy in order to improve the productivity of such application programs. We have developed the tool, called *User Interface Design Tool*, that facilitates to design and modify the objects and to examine them on the screen independently of application programs. This paper describes the structure and functions of the tool.

1. はじめに

高機能ワークステーション上のアプリケーション・プログラム（ＡＰ）では、ユーザインタフェース（ＵＩＦ）としてウィンドウやメニューなど（ＵＩＦ部品と総称する）を駆使した視覚型ユーザインタフェースを採用し、ユーザの使い勝手の向上を図ったものが多い。しかし、このようなＵＩＦの設計／開発には多大な工数を要し、それがＡＰ全体の開発効率向上に対する障害となっている〔１〕。

この障害の一例として、上記ＵＩＦ部品の設計／開発／保守に関する問題がある〔２〕。例えば、ＵＩＦ部品の表示形態や表示順序は、ＡＰ開発の初期の段階で定めたものが最良とは限らず、ＡＰ開発途中や開発完了後に変更が発生しがちなものである。一般に、これらの変更はＡＰ本体（ＡＰ本来の機能を実行する部分）の再コンパイル／リンクを必要とする場合が多く、容易には行えない。

今回、この問題を解決する方法として、ＵＩＦ部品をＡＰ本体から切り離して定義し、その表示形態や表示順序の設計／変更をＡＰ本体とは独立に行えるツール（ＵＩＦ設計ツール）を当社エンジニアリングワークステーションＭＥ１０００シリーズ上に開発した。

本稿では、このＵＩＦ設計ツールの構成と機能について述べる。

2. ＵＩＦ部品

ＭＥ１０００シリーズ上の対話型ＡＰでは、ユーザとの対話手段として、ウィンドウ、メニューおよびフォームが多く用いられる。これらを総称してＵＩＦ部品と呼ぶ。

2. 1 ウィンドウ

ＡＰの処理結果（文字、図形およびイメージ）の出力や、ＡＰへのデータ入力（ポインティング・データおよびキーボード・データ）を行うための表示画面上の矩形領域である。形態的には、ネーム・エリア、スクロール・バーおよびユーザ定義ボタンから成る操作枠を持つ「枠ありウィンドウ」と、それらを持たない「枠なしウィンドウ」とに大別される（図１）。

2. 2 メニュー

枠なしウィンドウの一種であり、形態的には、同ウィンドウを直線で区切られた１個以上の小区画（アイテム）に分割したものである。ユーザは所望のアイテムをマウスでクリックすることによりコマンド／データの入力を行う。ユーザの操作方法に関

して、通常メニューとポップアップ・メニューの二種類があり、後者の場合には、あるアイテムにマウス・カーソルを置くと、そのアイテムに付随するメニューが現われる階層型メニューが実現可能である（図２）。

2. 3 フォーム

枠なしまたは枠ありウィンドウの一種であり、同ウィンドウ内にソフト・スイッチ（アイテム）や文字列入力領域が配置されたものである（図３）。ユーザはソフト・スイッチをマウスで選択したり、文字列入力領域にキーボードから文字を入力することにより、ＡＰに対してデータ入力を行う。

ＭＥ１０００シリーズには、ＡＰからの指示に従ってこれらＵＩＦ部品の表示を行い、ユーザからの入力を受け付けるライブラリ群（以下、既存ライブラリ群と呼ぶ）が用意されている〔３、４〕。

3. ツール開発の背景と目的

ＭＥ１０００シリーズ上の対話型ＡＰにおいて上記ライブラリ群を利用するためには、ＵＩＦ部品を定義するデータ（例えば、ウィンドウの表示位置、サイズおよび属性を定義するデータであり、以下では単に定義データと称する）をソース・プログラム中に数多く用意する必要がある。一方、個々のＵＩＦ部品の形態や配置などは、ＡＰの仕様設計段階で決定したものが最良のものとは限らず、ＡＰ開発途中（または、開発完了後）での修正／変更が発生しやすいものである。これは、最良のＵＩＦの設計指針を明確に定めることが極めて困難なことに起因する。従って、これらの修正／変更は必然的に発生するものであり、ＵＩＦ部品の設計／開発／保守の効率を改善することは、対話型ＡＰの開発効率を向上させるために不可欠である。

以上のような背景から、以下の４つの目的を持って、ＵＩＦ設計ツールの開発を行った。

(1) レイアウトなどの早期確認

ＡＰの使い勝手の観点からは、ＵＩＦ部品のレイアウト（位置、サイズ）や表示順序も重要な問題である。ＡＰの開発効率を向上させるためには、これらをＡＰ開発の初期の段階で、実際に表示画面上で確認できることが必要である。

(2) 確認結果の、ＡＰにおける直接利用

ＡＰ開発者が、上記(1)で確認された結果を参照しながらＵＩＦ部品の設計／開発を行うのであれ

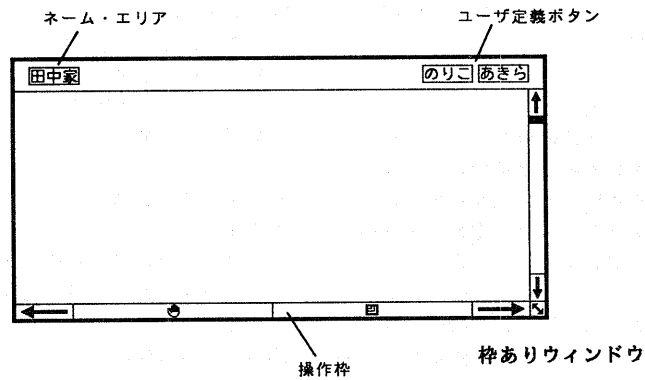


図 1. ウィンドウ

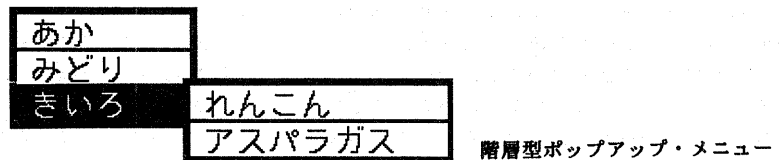


図 2. メニュー

ネームエリア・ユーザボタン

取消 終了

ネームエリアを指定しますか。 ☐ Yes ☐ No

ネームエリア名: 田中家

ユーザボタン数を選択して下さい。

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ なし

ユーザボタン1: あきら シグナル番号: -01

ユーザボタン2: のりこ シグナル番号: -01

文字列入力領域

ソフト・スイッチ

図 3. フォーム

ば、AP開発の効率向上はあまり期待できない。即ち、上記(1)で確認された結果をAP本体で直接利用できることが必要である。

(3)修正/変更結果の即時反映

AP開発途上に、UIF部品のレイアウトなどの修正/変更を要求される場合が多い。このような場合、AP本体（または、その一部）の再コンパイル/リンクを必要とせず、極力定義データの修正/変更だけで対処できることが望ましい。

例えば、メニュー・アイテムの変更（「コピー」を「転写」に変更、など）だけの場合には、AP本体の処理ロジックを変える必要がないため、定義データの変更のみで対処できる。

(4)定義データの対話型入力

現状では、定義データをハンドコーディングにより記述するのが通常である。一方、定義データは上記ライブラリ群が提供する基本機能に対応して記述されるため、一般に記述レベルが低く、細部に渡る定義まで記述しなければならない。従って、必然的にデータ記述量が多く、また記述内容が複雑であり、AP開発者の負担となる。そのため、AP開発者と対話しながら定義データを入力できるようなツールが必要であり、そこでは必要不可欠なデータのみを入力させるような工夫も必要である。

4. ツールの構成と機能概要

4.1 ソフトウェア構成と概略機能

UIF設計ツールはUIFedt（UIFエディタ）、UIFtrn（UIFトランスレータ）およびUIFlib（UIFライブラリ）の3つのソフトウェア・モジュールから構成される。これらのモジュールと、APおよび既存ライブラリ群との関係を図4に示す。UIF設計ツールは、既存ライブラリ群に全く手を加えず、それに外部から付加する形で実現されている。

(1)UIFedt

ユーザが定義データを作成するために、対話的な入力/編集を行うモジュールである。入力されたデータは、ディレクトリINTRの下の中間ファイルxxxx.intに格納される。また、制御ファイルCN TFの生成と管理も行う。

(2)UIFtrn

UIFedtで作成した中間ファイルxxxx.intをUIFlibが利用可能な形式に変換し、ディレクトリEXECの下の実行形式ファイルxxxx.exeを生成するモジュールである。1つのUIF部品の設計

完了時に、UIFedtより呼び出される。

(3)UIFlib

APからの指示に従って、必要なxxxx.exeを読み出し、既存ライブラリ群に適合する形式にアセンブルした後、既存ライブラリ群に処理を依頼するモジュールである。

4.2 データ構造

(1)CNTF

UIF部品名、定義データ名および描画コマンド・ファイル名の対応関係を保持するファイルである。

(2)INTR/xxxx.int

ユーザが入力したデータを、UIFedtが扱いやすい形式で保持するファイルである。各定義データ毎に一つ存在する。

(3)EXEC/xxxx.exe

既存ライブラリ群へ与えるデータを保持するファイルである。xxxx.intと一対一に対応している。内部形式はC言語のオブジェクト形式と同様である。

(4)PICT/yyyy.com

描画コマンド群を保持するファイルである。CNTFにおいて、あるウィンドウにこのファイルが指定されていると、同ウィンドウが表示される時に同ファイルで指定された描画が行われる。

以上の各データの関係を図5に示す。図5は本ツールに3つのウィンドウ（識別名がaaaa,bbbb,cccc）が登録され、ウィンドウaaaaとbbbbが同一定義データを使用している状態を表わしている。

5. 各UIF部品の設計概要

5.1 ウィンドウの設計

ウィンドウの設計は図6に示す「ウィンドウ編集フォーム」を中心に行なわれる。同画面はウィンドウの各種属性を設定するためのものであり、必要に応じて、属性値を設定するためのメニューやフォームがオーバーラップ表示される。

ウィンドウの位置およびサイズの設定は実際に対象ウィンドウを表示し、それを直接操作することにより行う。その操作方法是ウィンドウ・システムが提供している「ウィンドウ操作」のものと同一である。さらに、ツールに登録されている複数のウィンドウを同時に表示しそれらを直接操作することにより、位置やサイズの関係の調節を行うことも可能である。

なお、ほとんどの操作はマウスのクリックで行えるようにし、キーボードの使用は必要最小限となる

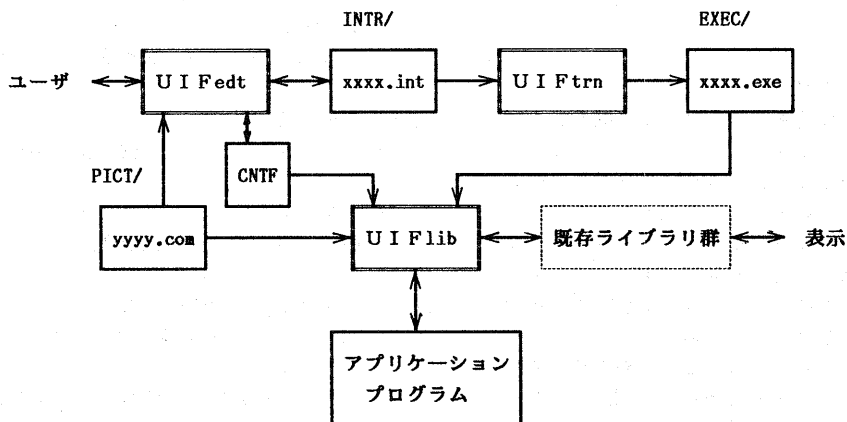


図4. UIF設計ツールの構成

CNTFの構造

識別名	定義データ名		描画コマンド群
aaaa	xxxx.int	xxxx.exe	yyyy.com
bbbb	xxxx.int	xxxx.exe	zzzz.com
cccc	xxxx.int	uuuu.exe	—

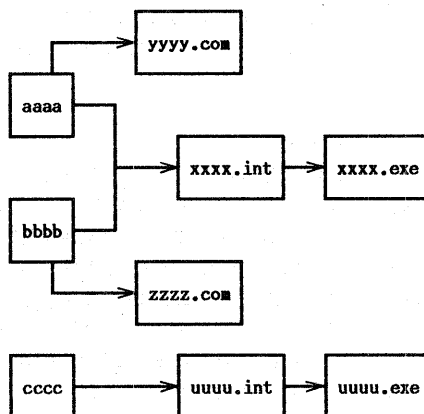


図5. 各データの関係

ウィンドウ編集フォーム
設定/修正 表示 取消 終了

選択されたウィンドウ名、定義データ名は以下の通りです。

ウィンドウ名: tanaka , 定義データ名: tanaka

コメント名:

操作枠の形式を指定して下さい。

☒ フル描画
 ☐ ページ間スクロールなし
 ☐ 操作枠なし

ウィンドウ種別を指定して下さい。

☒ グラフィック
 ☐ モックアップエミュレート
 ☐ GKS

表示クラスを指定して下さい。

☐ システム
 ☒ ユーザー1
 ☐ ユーザー2

ネームエリア及びユーザボタンを設定しますか。

枠の色及びアレーン枚数、また各アレーンの色設定しますか。

論理画面を設定しますか。

仮想画面の生成を行いますか。

図6. ウィンドウ編集フォーム

よう工夫した。

5.2 メニューの設計

メニューの設計は基本的にアイテムの個数と各アイテムの文字列を設定することにより行われる。メニューとなる枠なしウィンドウのサイズはアイテム数および文字列長から自動的に計算される。

階層構造を持つポップアップ・メニューの場合には、図7に示すようにその全体構成を常にユーザに提示し、現在設計中のメニューがどの位置のものかを容易に把握できるようにしている。

さらに、ポップアップ・メニューの確認は、実使用時の操作と同一の操作で行われる。即ち、図7のウィンドウ内でマウスの右ボタンを押しすることにより、マウス・カーソルの近傍にメニューが表示され、カーソルの移動に従ってアイテムの反転および下位メニューの表示が行われる。所望のアイテム内で右ボタンをリリースすることによりメニューが消去される。

5.3 フォームの設計

フォームの設計は、図8に示す、ガイドとなるグリッドが表示された空のフォームに対して、ソフト・スイッチ(アイテム)や文字列入力領域を直接配置していくことにより行われる。例えば、ソフト・スイッチ(アイテム)を表わす矩形領域は、その左上隅点と右下隅点をマウスでクリックすることにより設定される。その中の文字列は、対象矩形の近傍にオーバーラップ表示される文字列入力用フォームを用いて行われる。

フォームのサイズは、その中に配置されたソフト・スイッチ(アイテム)や文字列入力領域の状態により自動的に計算する方法と、ユーザからの指定により決定する方法の両方を用意している。

6. APとの結合

6.1 UIFlibの関数

APはUIFlibを通して、実行形式の定義データ(xxxx.exe)を利用する。この時、APは使用するUIF部品の識別名のみを知ればよく、同部品の詳細な定義データを知る必要はない。現在、UIFlibには、各UIF部品対応に以下のC関数を用意されている。

(1)ウィンドウの表示

utdisp_win (ウィンドウの識別名)

(2)ウィンドウの削除

utdel_win (ウィンドウの識別名)

(3)メニューからの入力

ut_menuin (メニュー識別名, X, Y)

(4)フォームの表示

utdisp_frm (フォーム識別名, 制御データ)

(5)フォームの削除

utdel_frm (フォーム識別名)

(6)フォームからの一括データ入力

ut_frmin (フォーム識別名, 入力データ)

なお、メニューの表示に関しては、その位置をAPから動的に制御できるようにするため、位置(X, Y)を指定するようにしている。

また、ウィンドウとフォームに関して、既存ライブラリ群には上記(1)~(6)では利用できない機能(ウィンドウ・サイズの動的変更やフォーム内容の動的編集など)が用意されている。APはウィンドウおよびフォームの表示関数(上記(1)および(4))のリターン値として与えられる識別子を用いて、このような既存ライブラリの機能を直接利用することができる。

6.2 UIFlibを利用しないAPとの結合

APが完成した時点では、移植性の観点からUIF設計ツールとは独立に、そのAPだけを単独に使用できることが望ましい場合も想定される。本ツールにおける実行形式ファイルにはC言語のオブジェクト形式を採用しているため、リンカを用いて容易にAPと定義データとの結合が行える。

さらに、ソース・レベルの定義データを必要とする場合も想定されるため、UIFedtの出力として定義データのインクルード・ファイルを作成することも検討している。

7. 考察

UIF設計ツールを用いたAP開発を3つのステップに分け、各ステップにおける本ツールの有効性について考察する。

(1)第1ステップ

APの仕様設計段階であり、未だAP本体(または、その一部)は存在しない。ここでは、UIF設計ツールを単独で用いて、そのAPで必要となる実行形式定義データの作成、およびUIF部品の表示レイアウトや表示順序の確認を行うことができる。その結果は、即座にAP仕様にフィードバックされる。

ポップアップメニュー修正 表示 閉鎖 編集 追加 取消 終了

入力メニュー名 : pop

ポップアップメニューは以下の構造となっています。
修正したいメニューを選んで下さい。

あか

みどり

さいろ

れんこん

アスパラガス

図 7. メニュー構造表示

ネームエリア・ユーザボタン おわり とりけし

ネームエリアを指定しますか? はい

アイテム設定 取消 終了

アイテム番号 1

表示文字列 はい

左詰 センタリンク 右詰

線 あり なし

選択時に反転 する しない

初期状態 反転 非反転

図 8. フォーム設計画面

但し、この段階では、当然のことながら、APからの表示は行えない。

(2)第2ステップ

AP本体の開発途上の段階であり、AP本体の一部とUIF設計ツールが連動してUIF部品の表示を行っている。ここでは、APからのデータ表示が一部稼働しており、その結果を見て、UIF部品の仕様を変更する必要がある。しかし、本ツールが上記形態で動作しているため、その変更にはAP本体を変更することなく容易に対処できる。

(3)第3ステップ

APの開発が一応完了した後の段階である。ここでは、APを使い込んだ結果、UIF部品の仕様変更が発生する。この時も、APのセマンティクスに影響を与えない限り、本ツールを用いて容易に対処することができる。例えば、同APがUIFlibを使用する形態であれば、UIFedtを用いて該当するUIF部品の実行形式定義データを変更するだけでよい。また、UIFlibを使用しない形態の場合には、実行形式定義データを変更した後、再度リンクを行えばよい。

このように、UIF設計ツールはAP開発の全般に渡って、その効率向上に寄与するものと考えられる。

8. おわりに

以上、UIF設計ツールの構成と機能について述べた。本ツールを用いることにより、UIF部品の定義データの作成効率が向上するとともに、APにおけるプログラミングの負担を軽減することが可能となった。

今後の課題として、

(1)UIFedtのユーザインタフェースの評価と改良

(2)登録された定義データを利用したユーザインタフェースの試行環境の整備

を検討する予定である。特に後者に関しては、現在UIF部品の表示やそれらに対する入力を行うUNIXコマンドを用意しており、シェル・プログラミングを行うことにより簡単な試行が行える。今後それらを発展させ、計算機の非専門家でも使えるものにしていく必要がある。

参考文献

- [1] H.Lieberman:There's More to Menu Systems Than Meets the Screen, SIGGRAPH'85, pp.181-189(1985).
- [2] 神原 他:ウィンドウのためのユーザインタフェース定義ツール, 情報処理学会第33回全国大会 7F-5(1986).
- [3] 三菱マルチメディア・エンジニアリング・ステーション ME1000シリーズ" ウィンドウ管理説明書, 三菱電機㈱ (1987).
- [4] 三菱マルチメディア・エンジニアリング・ステーション ME1000シリーズ" 操作支援説明書, 三菱電機㈱ (1987).