

ES/KERNEL/Wのユーザーインタフェース構築ツール「UIビルダ」

UI Builder: User Interface Development Tool Using ES/KERNEL/W

専門家の知識を表現して作るエキスパートシステムでも、実用システムとするためには、エンドユーザーインタフェースが必要である。しかし、ユーザーインタフェースを構築するための開発工数が膨大で、知識処理部よりも時間がかかるという問題があった。

UIビルダは、エキスパートシステム構築ツールES/KERNEL/Wを利用したエキスパートシステムのユーザーインタフェース開発を容易に行うためのツールである。UIビルダには、ユーザーインタフェースを対話的に定義する機構が備えられている。定義されたユーザーインタフェースを、知識表現言語からフレーム(UIフレーム)とみなして動作させることができる。ユーザー入力は、UIフレームへ自動的にメッセージ送信されてくる。

この結果、画面作成用のプログラムは不要になり、ユーザー入力を取り扱うプログラムも簡単に作成することができる。ウインドウシステムをそのまま用いて開発したユーザーインタフェースを、UIビルダを用いて作り直したところ、従来に比べてその開発工数が $\frac{1}{10}$ 以下になる見通しを得た。

増石哲也* *Tetsuya Masuishi*
小六正修** *Masanobu Koroku*
安崎篤郎*** *Tokurô Anzaki*
磯辺 寛*** *Hiroshi Isobe*

1 緒 言

専門家の知識を知識ベース化して実現するエキスパートシステムを実用システムとするためには、知識処理部分以外にエンドユーザーインタフェースが必要である。特に、ビジネス向けの場合は、対話形のエキスパートシステムになるケースが多いため、エンドユーザーにとって使いやすい対話用ユーザーインタフェースを開発することがシステムを実用化する上でのキーポイントになる。

近年、マルチウインドウやマウスを持つワークステーションが普及し、より柔軟なユーザーインタフェースを持つシステムを構築することが可能となってきた。マウスによる命令入力、ポップアップメニュー、アイコン、数値の右詰め入力などは、ワークステーション上のアプリケーションシステムにとって必要不可欠な機能である。エキスパートシステムでも、これらの高度なユーザーインタフェース^{1),2)}が要求される。

エキスパートシステム構築ツールを用いてエキスパートシステムを開発するとき、高度なユーザーインタフェースを実現しようとする、ユーザーインタフェース部分の開発工数が知識処理部分の開発工数よりも大きくなってしまいう問題がある。これは、エキスパートシステム構築ツールが具

備する知識表現機能によって、知識処理部分を簡単に開発できるのに対して、ユーザーインタフェース部に関しては従来どおりのプログラミングを必要としていたことに起因する。

本論文で述べるES/KERNEL/W (Expert System/KERNEL/Workstation system)-UI (User Interface)ビルダは、この問題を解決するために開発されたユーザーインタフェース構築ツールである。特に、ビジネス向けアプリケーションシステム用のユーザーインタフェースを提供している。記述実験の結果、UIビルダを用いることによって、ES/KERNEL/Wで開発するエキスパートシステムのユーザーインタフェース部分の開発工数が、従来に比べて $\frac{1}{10}$ 以下になる見通しを得た。

2 エキスパートシステム構築ツールでの位置づけ

エキスパートシステムは、専門家の知識をエンドユーザーに利用させるためのメディアとして働くシステムである。したがって、知識を蓄積するための知識ベースと、知識を解釈するための推論機構のほかに、エキスパートが知識を入力するためのエキスパート用ユーザーインタフェースと、エンド

* 日立製作所システム開発研究所 ** 日立製作所情報事業本部 *** 日立製作所ソフトウェア工場

ユーザーがエキスパートシステムを利用するためのエンドユーザーインターフェースが必要となる。

エキスパートシステム構築ツールとしては、これらエキスパートシステムの各コンポーネントの構築を支援するための機能が備わっていなければならない。UIビルダは、エキスパートシステム構築ツールES/KERNEL/Wで、エンドユーザーインターフェースの構築支援を目的としたツールとして位置づけられる³⁾。

3 UIビルダが提供するユーザーインターフェース

3.1 パネル機能

ビジネス向けのエキスパートシステムには、エンドユーザーとエキスパートシステムが対話するための画面が不可欠である。対話用画面には、単に画面表示だけでなく、マウス入力の読み込み・カーソル移動などの複雑なプログラミングが必要となる。相続相談エキスパートシステム¹⁾のユーザーインターフェース開発にかかった工数の大部分が、この複雑なプログラムのデバッグに必要となったものである。

パネルは、対話用の画面を作成するための機能である。パネルが提供する対話用のユーザーインターフェースとして以下のものがある(図1参照)。

(1) 入力域アイテム

文字列や数値をキーボード入力する領域である。数値は右詰めされる。

(2) 出力域アイテム

文字列・数値などを表示する領域である。数値は右詰め表示される。文字列形の出力域アイテムは、メニューとして用いることもできる。

(3) 選択肢アイテム

どれかひとつを選択するメニューである。

(4) アイコンアイテム

形状を持つアイコンである。

(5) ボード

上記の各種アイテムを複数個まとめて取り扱うためのものである。ボードに接続された各種アイテムは、ひとまとまりに表示・消去することなどができる。

(6) ポップアップ

マウスでピックすると現れるメニューである。

3.2 ビジネスグラフ機能

融資エキスパートシステム⁴⁾では、エキスパートシステムの質問に対してユーザーが適切に答えられるように、融資先の業績データなどをグラフ表示するユーザーインターフェースを持っている。また、推論結果をグラフ化して表示したいというニーズもある。

UIビルダを用いると、フレームデータを棒グラフ・円グラフ・レーダチャートなどのビジネスグラフとして表示する機能を簡単に実現することができる。グラフの種類、軸の幅・目盛り、タイトルの位置・大きさなどを対話的に定義し、推論実行時にフレームデータをその形式に合わせてグラフ化する。

3.3 ドキュメント機能

プログラミングノウハウ伝承システムSOCKS (Software Common Knowledge Transfer System)⁵⁾は、ベテランプログラマのノウハウをワードプロセッサ文書としてあらかじめ格納しておき、質問応答の結果に応じて特定のページを表示するというユーザーインターフェースを持っている。

また、融資エキスパートシステムでは、グラフ表示で対話時のユーザーを支援したが、文書表示でも同様に支援することができる。特に、イメージスキャナで読み込んだ画像を表示すると効果的である。

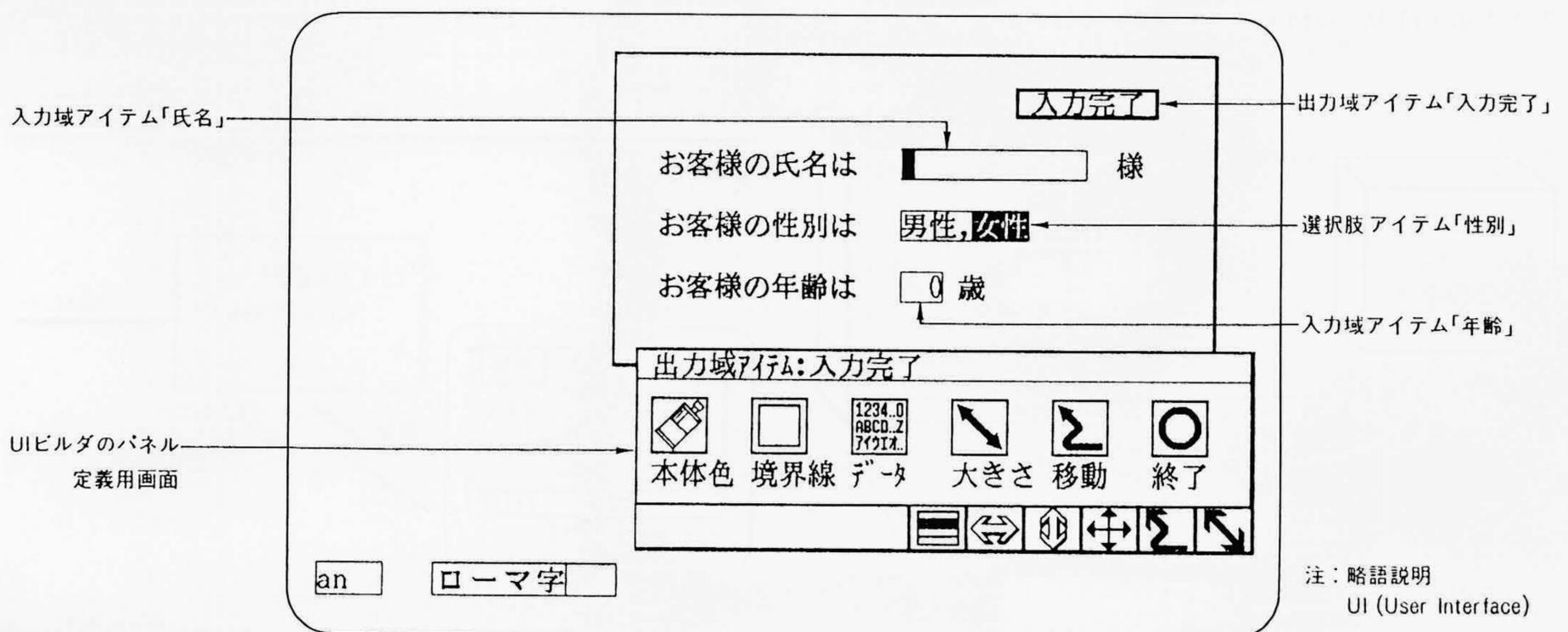


図1 パネルとパネル定義 UIビルダは、推論実行時に利用される画面を対話的に定義することができる。これによって、画面作成用のプログラムが不要になる。また、実際に利用される画面そのものを確認しながら画面設計を行うことができる。この図はパネルを作成する状態を示したもので、入力域アイテム、出力域アイテム、選択肢アイテムなどが並べられている。

UIビルダは、OFIS(Office Automation and Intelligence Support Software)シリーズで作成した文書を推論実行時にページ単位で表示する機能を提供している。これにより、SOCKSで実現した機能や質問応答時にドキュメント表示してユーザーを支援する機能を、簡単に実現することができる。

4 UIビルダとUIフレーム

4.1 UIビルダとUIオブジェクト

UIビルダは、2章で述べたユーザーインタフェースを対話的に定義するツールである。これにより、画面作りのためのプログラミングは不要になる。また、エキスパートシステムで実際に用いられる画面そのものを確認しながら、ユーザーインタフェースを作成することができる。

UIビルダで作成されたユーザーインタフェースは、それぞれ名前を付けられ、ファイルに格納される。このデータをUIオブジェクトと呼ぶ。推論実行時に、UIオブジェクトをロードして利用する(図2参照)。

4.2 UIフレーム

ES/KERNEL/Wで作るエキスパートシステムでUIオブジェクトを利用するためには、ES/KERNEL/Wの知識表現言語からUIオブジェクトを操作するための言語インタフェースが必要である。

UIオブジェクトは、知識表現言語からはフレームとして見える。UIオブジェクトを表現するフレームをUIフレームと呼ぶ。UIフレームを作るために、ES/KERNEL/WはUIフレーム用のクラスフレームを提供する。提供されたクラスフレームを継承するインスタンスフレームを定義することによってUIフレームを作ることができる。

提供するクラスフレーム(表1参照)には、UIオブジェクトを操作するためのメソッドが用意されている。このメソッドには、位置・大きさ・色などの属性を設定するもの、属性を参照するもの、UIオブジェクトを動的に生成するものなどがある。UIフレームに対して、メッセージ送信することによって、表示・消去、各種属性変更など、UIオブジェクトに対す

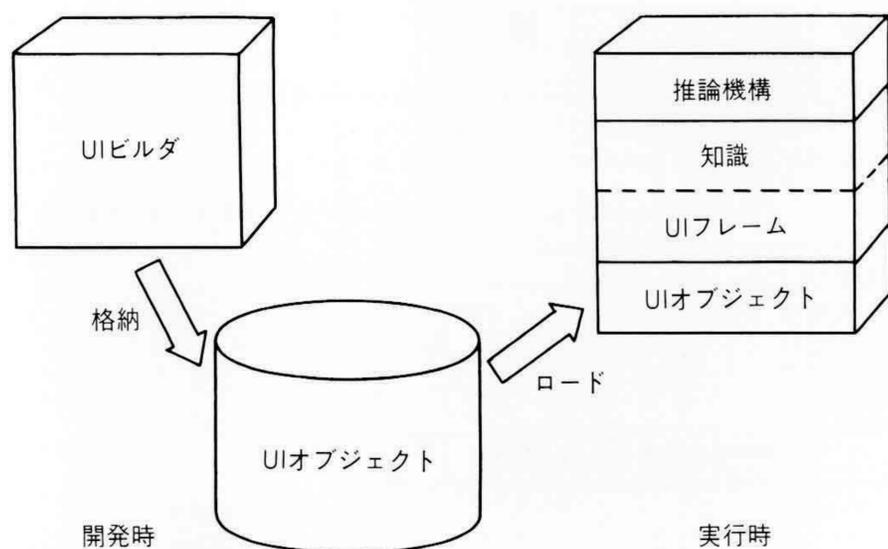


図2 UIビルダとUIフレーム UIビルダで対話的に編集した画面データは、推論実行時にローディングされる。表示や読み込みなどの処理を、UIフレームに対するメッセージ送信によって行うことができる。

る操作をオブジェクト指向形言語的に行うことができる(図3参照)。

4.3 パネルにおけるユーザー入力読み込み

パネルは対話用のユーザーインタフェースなので、ユーザーからの入力(マウスピック・ポップアップの選択・キーボード入力など)を受け付けなければならない。これを言語インタフェースでどのように取り扱うかが問題となる。

ユーザーからパネルへの入力は、ユーザーからUIフレームへのメッセージ送信として解釈する。例えば、ある出力域アイテムがマウスでピックされたとき、ピックされたことを報告するメッセージがUIフレームに送信されてくる(図4参照)。このメッセージによって、UIフレームのアクションメソッドが起動される。アクションメソッドは、UIオブジェクトに対する操作と一対一で対応するメソッド名を持ち、引き数の仕様もそれぞれ決められている(表2参照)。この仕様に合わせてアクションメソッドを記述すれば、ユーザーの操作に対応した動作を指定することができる。

表1 提供フレーム UIビルダの提供する各機能に対応して、UIフレーム用のクラスフレームを提供している。この提供フレームを継承したインスタンスフレームを作成することによって、UIビルダで作った画面を推論実行時に利用することができる。

提供フレーム	用途
#ウィンドウ	パネルウィンドウ用
#入力域アイテム	各種パネルの ・表示・消去 ・属性設定 ・属性参照 ・動的生成・削除 ・デフォルトアクション
#出力域アイテム	
#選択肢アイテム	
#アイコンアイテム	
#ボード	
#ポップアップ	
#グラフ	ビジネスグラフの表示・消去
#表ビュー	フレームデータのグラフ化
#チャート定義	グラフ構造の動的設定
#ドキュメント	ドキュメントの表示・消去

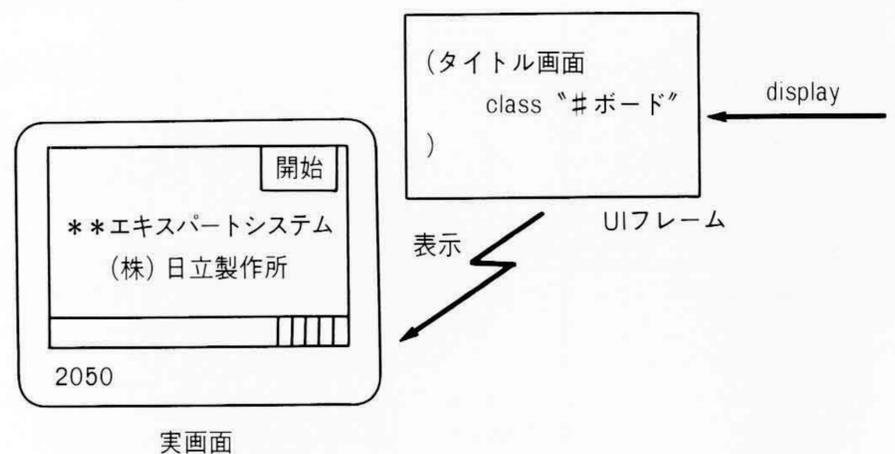


図3 UIフレームの働き UIフレームにメッセージ送信することによって、表示や消去、更に色・位置・大きさ・データなどの各種属性を設定することができる。UIビルダで作られていないものを、推論実行過程で動的に生成することもできる。

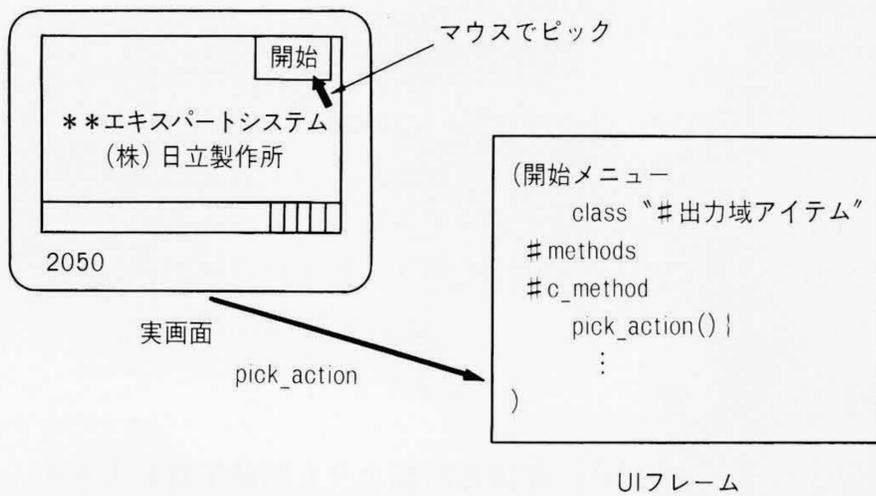


図4 ユーザー入力の読み込み パネルに対するユーザーの入力は、UIフレームに対するメッセージで報告される。このメッセージにより、各種アクションメソッド(表2)が起動される。

表2 アクションメソッド アクションメソッドは、ユーザーの入力を自動的に報告するメソッドである。提供するUIフレーム用のクラスフレームには、これらのアクションメソッドの標準機能が作り込まれている。パネルに対応するUIフレームでアクションメソッドを記述しなければ、標準機能が動作する。

アクションメソッド	機能
pick_action	マウスでピックされたことを報告
select_action	選択肢の選択文字列と選択番号を報告
popup_action	ポップアップの選択文字列と選択番号を報告
input_action	入力文字列を報告
pf_key_action	PFキーが押下されたことを報告

提供されるフレーム用のクラスフレームには、あらかじめ典型的なアクションメソッドが提供されている。これをデフォルトアクションと呼ぶ。デフォルトアクションが提供する機能で不十分な場合は、UIフレームでアクションメソッドを書き換えればよい。

先の図1に示したパネルからの入力データを「入力データ」というフレームに代入する記述例を示したものが図5である。入力域アイテム「氏名」の入力値を「氏名」というスロットに、選択肢アイテム「性別」で選択された文字列を「性別」というスロットに、入力域アイテム「年齢」の入力値を「年齢」というスロットに入れる。特にUIフレーム「年齢」は、ES/KERNEL/W提供関数“upk_frck”を用いてスロットの値域(range)を満足するかどうかをチェックする機能を実現している。

カーソルキーやマウスによるカーソル移動、入力域の中の挿入・削除、順不同の入力、数値の右詰め、マウスでピックしたときの動作、入力データのチェック・代入など従来は複雑なプログラミングを必要としていた機能が、図5に示したプログラムで実現できる。なおここでは、マウスを用いたユーザーインターフェース(「入力完了」をピックすると入力を完了する。)を例としたが、キーボードだけで入力するユーザーインターフェースも同様に実現可能である。

```

(入力データ
class system
  氏名 (data_type string)
  性別 (data_type string)
  年齢 (range [0 150])
)
(ウィンドウ
class "#ウィンドウ"
)
(入力完了
class "#出力域アイテム"
#methods
#c_method
  pick_action() {
    :
  }
)
(氏名
class "#入力域アイテム"
マッピングフレーム 入力データ
マッピングスロット 氏名
)
(性別
class "#選択肢アイテム"
マッピングフレーム 入力データ
マッピングスロット 性別
)
(年齢
class "#入力域アイテム"
マッピングフレーム 入力データ
マッピングスロット 年齢
#methods
#c_method
  input_action(age) int age;
  {
    if(upk_frck("入力データ","年齢",age)){
      send("年齢","put_color",1);
      send("ウィンドウ","reactivate_panel");
    }else{
      send("年齢","put_color",7);
      send("年齢","reserve_mapping",age);
    }
  }
#methods_end
)

```

図5 UIフレームの記述例 入力域アイテムの入力データ及び選択肢アイテムの選択文字列を、特定のフレームのスロットに値を代入するためには、「マッピングフレーム」と「マッピングスロット」にフレーム名とスロット名を指定するだけでよい。この動作を更にカスタマイズするためには、ユーザー入力を報告するアクションメソッドを書き換える。この例では、「年齢」という入力域アイテムの入力データが、フレーム「入力データ」のスロット「年齢」の値域(range)を満足しているかどうかをチェックするように、フレーム「年齢」のアクションメソッド“input_action”を書き直している。

5 効果

(1) ユーザーインターフェース開発の工数削減

UIビルダを用いることによって、ユーザーインターフェースの開発工数を大幅に削減することができる。C言語でウィンドウ関数を用いて記述したユーザーインターフェースをUIビルダで作り直した一例では、C言語記述部分が2.1kステップから0.2kステップに減った。

(2) 宣言的プログラムによる高品質化

従来、特にユーザー入力の読み込みの部分に関して、ウィンドウシステムから得られるエスケープシーケンスを解析するなど複雑なプログラミングを必要としていたが、UIビルダを用いた場合「このアイテムがピックされたら、**しなさい」という宣言的なプログラムを記述すればよく、プログラミングミスがほとんど入り込まない。この結果、ステップ数で比較する以上の工数削減が期待できるほか、品質の高いユーザーインタフェースが構築できる。

6 エキスパートシステム開発に与える影響

UIビルダを利用することによる工数削減は、エキスパートシステム開発に様々な影響を与える。

(1) プロトタイピングが簡単にできる。

エキスパートシステムは、システム開発部門以外に所属する「専門家」の知識によって成り立っている。知識を収集するために断続的なインタビューを行いながら開発を進めることが多い。このため、プロトタイピングによって仕様を決定していくことが肝要である。

従来は、ユーザーインタフェース部の開発工数が大きいため、プロトタイプシステムでは簡単なユーザーインタフェースしか作らないことが多かった。しかし、計算機に手慣れない専門家やエンドユーザーにプロトタイプシステムを試用してもらうためには、実用システムレベルのユーザーインタフェースが必要である。

UIビルダを用いれば、ユーザーインタフェースを含めたプロトタイプシステムを、早い時期に開発することができ、専門家やエンドユーザーに試用してもらうことによって仕様設計を円滑に進めることができる。

(2) 知識ベース設計に注力できる。

エキスパートシステムは、その知識の質によってシステムの良しあしが決定される。したがって、知識獲得を含めた知識ベース設計が何よりも重要である。

UIビルダを用いると、ユーザーインタフェースの設計やデバッグに煩わされることなく、知識ベースの設計に注力することができる。

(3) 評価・修正のサイクルをスピードアップできる。

エキスパートシステムは、その処理内容を専門家の知識に頼っているため、システムの評価も専門家に頼らざるを得な

いことがある。処理結果が適切かどうか、入力項目が十分かどうかなどのチェックを専門家が行い、実際の修正をシステム開発部門で行うことになる。このため、評価から修正に至るまでに時間がかかる可能性がある。

UIビルダを用いると、ユーザーインタフェース部分を簡単に修正でき、評価・修正のサイクルをスピードアップすることができる。

7 結 言

UIビルダは、従来、知識処理部よりも開発工数が大きかったユーザーインタフェース部分を、簡単に作成するためのツールである。提供するユーザーインタフェースとして、対話用画面のパネル、フレームデータのグラフ表示、OFISシリーズで作成した文書の表示を用意しており、これらの機能を簡単にエキスパートシステムのユーザーインタフェースとして実現することができる。

対話的に作成されたユーザーインタフェースは、UIフレームを通じてエキスパートシステムの一部として動作させることができる。パネルへの入力、UIフレームへのメッセージとして送信されてくる。パネルへの操作に対応するアクションメソッドを記述すれば、ユーザー入力を簡単に扱うことができる。

実用化の時期を迎えたエキスパートシステムにとって、ユーザーインタフェース部分の開発工数を削減することは重要な課題であった。UIビルダは、この問題を解決したツールであると考えている。

参考文献

- 1) 中村, 外: 相続相談エキスパートシステム, 日立評論, 70, 11, 1150~1155(昭63-11)
- 2) 小暮, 外: 工場防災診断エキスパートシステム, 日立評論, 70, 11, 1166~1170(昭63-11)
- 3) 吉村, 外: ES/KERNEL/Wの開発環境, 日立評論, 70, 11, 1094~1099(昭63-11)
- 4) 森, 外: 金融機関におけるエキスパートシステム, 日立評論, 69, 3, 255~258(昭62-3)
- 5) 辻, 外: ノウハウの知識ベース化によるシステム・ソフトウェアの品質管理, 品質, 18, 2, pp.123~129(昭63-4)