U.D.C. 〔681.32.06:159.95〕.002.2:681.327.13

ES/KERNEL/Wの開発環境

Expert System Building Environment for ES/KERNEL/W

吉村紀久雄*　*Kikuo Yoshimura*
増石哲也**　*Tetsuya Masuishi*
松井　隆*　*Takashi Matsui*
塙　信弘*　*Nobuhiro Hanawa*

エキスパートシステムを開発するときには，エディタやデバッガなどのツール群が必要になる。エキスパートシステム構築ツールとして，知識表現言語や推論機構のほかにこれらのツール群を具備しなくてはならない。エキスパートシステム構築ツールES/KERNEL/Wでは，エキスパートシステムの構造モデルを新たに定義し，ツール群をこのモデルに沿って用意している。このモデルでは，専門家自身が知識を追加・修正するためのインタフェースを設けている。

構文誘導形の知識エディタを提供し，ES/KERNEL/Wの文法に精通していなくても知識を入力・編集できる。知識の静的解析には知識ブラウザ，実行しながらのデバッグには知識デバッガを提供している。知識テーラは，計算機に手慣れない専門家自身が，知識を追加・修正することを可能にしている。また，実用システムに必す(須)であるエンドユーザーインタフェースを構築するために，UIビルダを用意して開発工数の削減を図っている。

1 緒　言

エキスパートシステムの開発には，システム開発部門以外の「専門家」が密接に参画するという特徴がある。専門家がシステム開発に参加することによって必要となる機能が，エキスパートシステム構築ツールには具備されていなければならない。例えば，専門家自身がエキスパートシステムの動作確認を行うことがある。したがって，知識の微少修正や追加は，専門家自身の手でできることが望ましい。

エキスパートシステム構築ツールES/KERNEL/W(Expert System/KERNEL/Workstation system)の開発環境は，上述の「専門家自身が知識を追加修正するためのインタフェース」を含んだ，エキスパートシステムの構造モデルに基づいて設計されている。知識エディタは構文誘導形の構造エディタで，ナレッジエンジニアがES/KERNEL/Wの文法に精通していなくても，知識を入力・編集できるようにすることをねらいとしている。知識の静的解析には知識ブラウザ，実行しながらのデバッグには知識デバッガを提供しており，知識の正当性を検証する作業を支援する。知識テーラは，専門家自身が知識を追加修正するのに必要な専用エディタ(カスタマイズドエディタ)を生成するための知識獲得ツールである。また，実用システムに必すであるエンドユーザーインタフェースを構築するために，UI(User Interface)ビルダを用意して開発工数の削減を図っている。

2 エキスパートシステムの構造モデル

ES/KERNEL/Wの開発環境は，エキスパートシステムの構造モデルに沿って開発用ツール群を用意している。

エキスパートシステムの構造モデル(**図1**参照)では，3種のユーザーがエキスパートシステムに関与する。すなわち，知識を利用する人(エンドユーザー)，知識を保有する人(専門家)，知識を移植する人(ナレッジエンジニア)である。

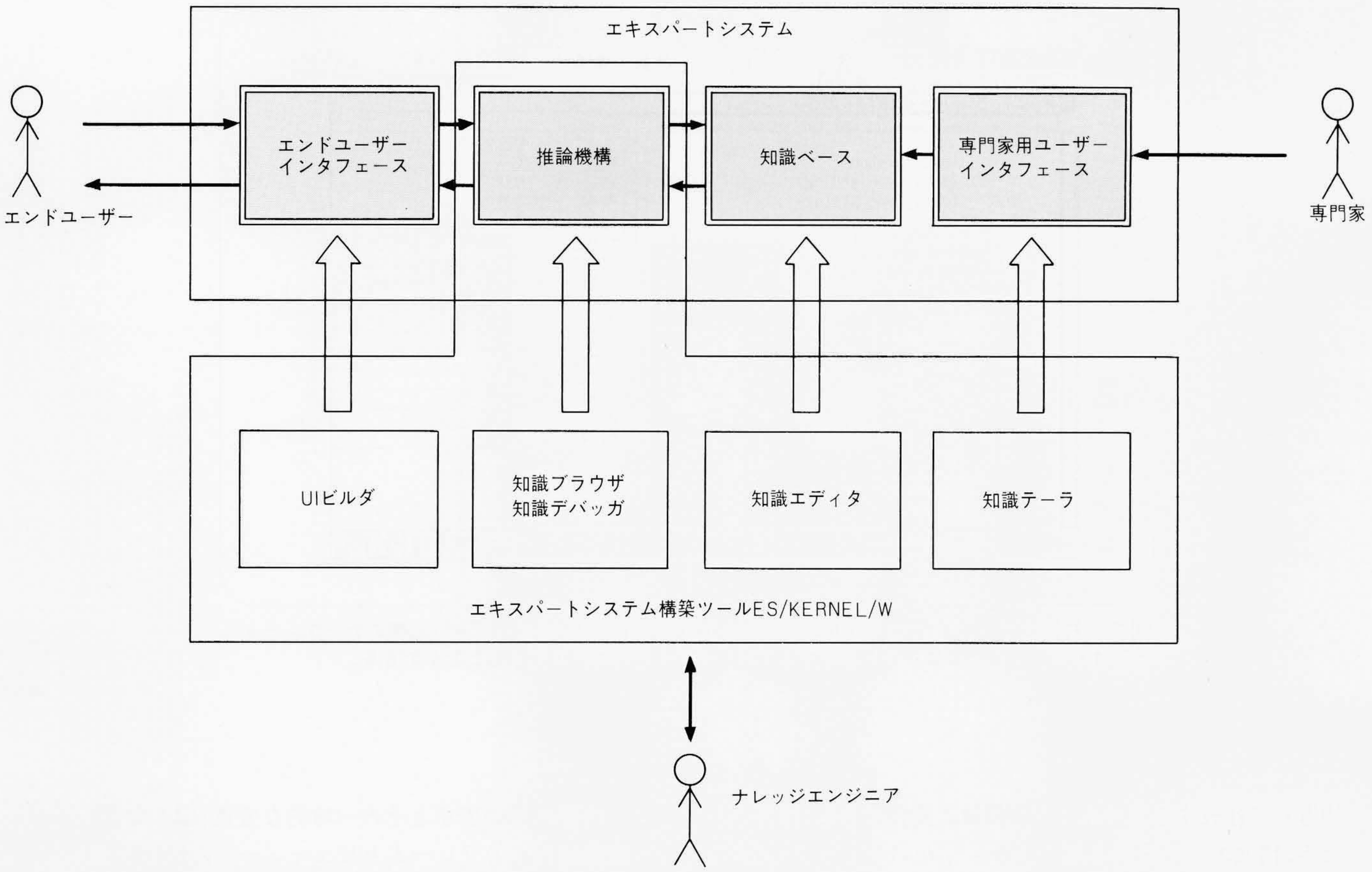
専門家自身がエキスパートシステムの知識を追加修正するためのインタフェースを持っていることが，このモデルの特徴である。このインタフェースを用意することによって，エキスパートシステムは，専門家の知識をエンドユーザーに伝承するためのメディアとして機能する。エンドユーザーと専門家に用いられるエキスパートシステムを，ナレッジエンジニアがエキスパートシステム構築ツールを用いて構築する。

このモデルでは，エキスパートシステムは次の4点のコンポーネントから構成されると考えている。

(1)　推論機構
(2)　知識ベース
(3)　専門家用ユーザーインタフェース
(4)　エンドユーザーインタフェース

図1のモデルでは，専門家自身が知識の微少修正や追加を行えるようにするために，上記(3)の専門家用ユーザーインタフェースをコンポーネントとして考慮している。(4)のエンド

* 日立製作所ソフトウェア工場　** 日立製作所システム開発研究所



注：略語説明　UI (User Interface), ES/KERNEL/W (Expert System/KERNEL/Workstation system)

図1　エキスパートシステムの構造モデル　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が利用するためのメディアとして機能する。このため，エキスパートシステムには，図示した4点のコンポーネントが必要となる。エキスパートシステム構築ツールには，この4点のコンポーネントそれぞれに支援機能が用意されている必要がある。

ユーザーインタフェースは，簡単な実験を行うためには推論機構が提供する標準画面で十分であり，特にコンポーネントとして取り上げられなかった。しかし，実際にエンドユーザーに利用される実用システムにするためには，使いやすいユーザーインタフェースを実現する必要があり，重要なコンポーネントである。

エキスパートシステム構築ツールES/KERNEL/Wは，エキスパートシステムの4点のコンポーネントに対して，それぞれ支援機能を用意した開発環境を提供している。

(1)　推論が適切に行われているかどうかを検証するためのデバッギングツール
　(a)　知識ブラウザ
　(b)　知識デバッガ
(2)　知識ベースを作るための開発ツール
　(a)　知識エディタ
(3)　専門家用のエディタを作るための知識獲得ツール
　(a)　知識テーラ
(4)　ユーザーインタフェースを簡単に開発するためのユーザーインタフェース構築ツール
　(a)　UIビルダ

更に，これらの支援機能を統合的に利用できるようにするために，開発環境制御機構がシステム全体を制御している。

3　知識の入力と編集(知識エディタ)

知識エディタは，ナレッジエンジニアが知識ベースを構築する作業を支援するために，知識入力の簡易化及び構文エラーの排除を実現している。

ES/KERNEL/Wの知識表現について，個々のキーワードや引き数の指定形式などを詳しく知らなくても，知識エディタの誘導に従い知識を入力・編集することができる。知識エディタは，カーソルの位置に文法上指定可能なキーワードや引き数の種類の一覧をテンプレートとしてウインドウに表示する(**図2**参照)。ナレッジエンジニアは，このテンプレートから必要なものをマウスで選択する。選択された項目がエディタウインドウ上に表示されていく。テンプレートによる誘導機能によって，キーボードを打つ回数が減り，間違いのない知識を短時間で入力することができる。

知識エディタは，入力された知識の構文エラーを排除する機能を持っている。キーボードで直接入力された知識を即座に構文解析し，エラーチェックしている。

構文の形式に着目した多彩な編集機能も用意している。例えば，カット アンド ペースト機能を用いて，知識の一部分

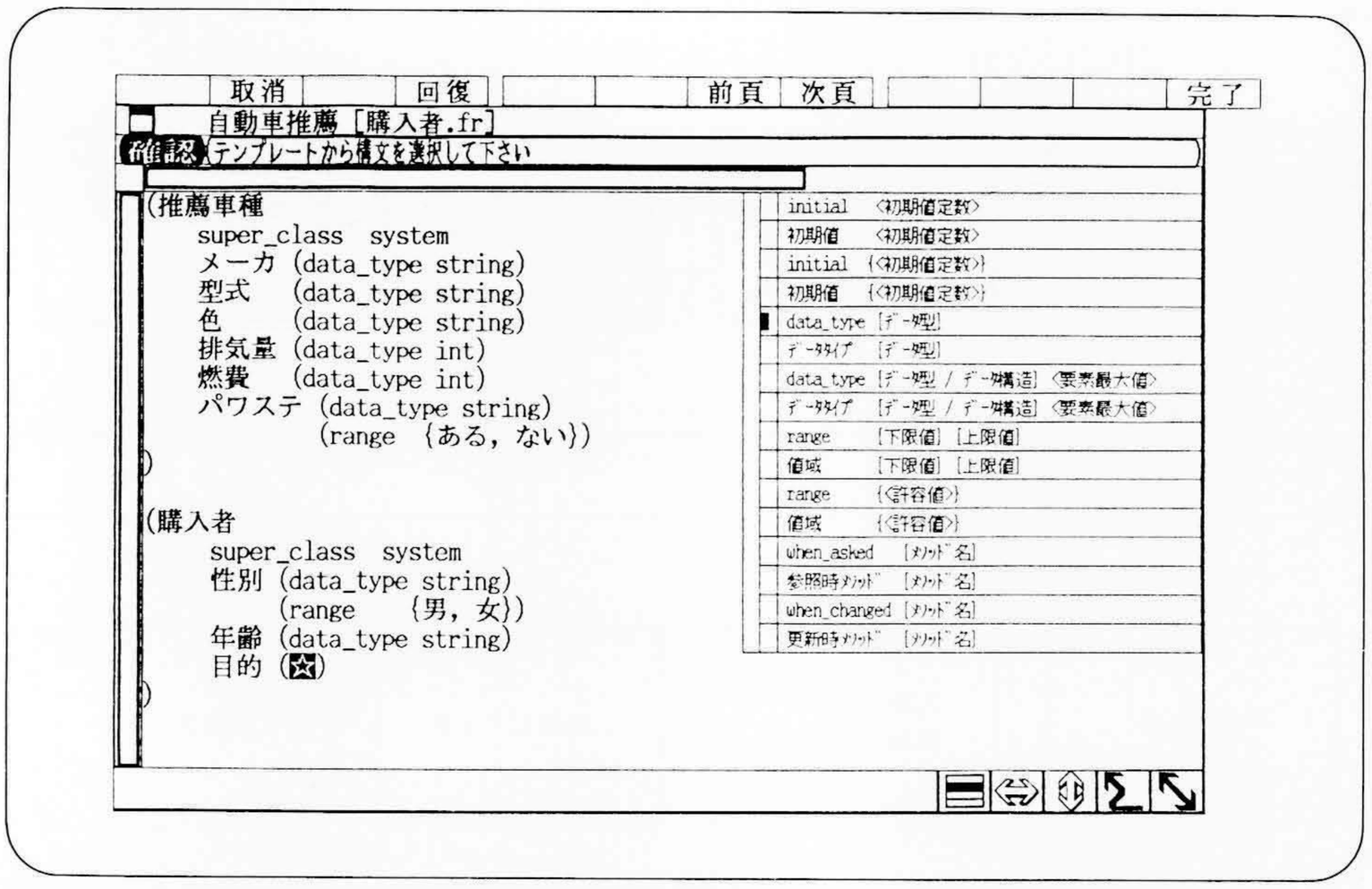


図2 知識エディタ カーソルの位置に指定可能なキーワードの一覧が，右側のテンプレートに表示される。このテンプレートをマウスでピックするだけで，入力することができる。

を複写して作成するようなことが簡単に実行できる。複写されたとき，構文エラーを引き起こすかどうかのチェックも行う。

4 知識の分析と試し実行(知識ブラウザ・知識デバッガ)

知識ベースが大規模になると，知識記述の誤り，不足，あるいは矛盾が生じやすくなるうえに，それらを検出することが困難になってくる。ES/KERNEL/Wは，知識の正当性検証を支援する環境として，知識ブラウザと知識デバッガを提供している。

4.1 知識の静的検証(知識ブラウザ)

知識ブラウザは，知識表現の枠組みであるフレーム，ビューノート，ルールなどの中で用いる名称の定義や参照関係を検証するためのツールである。

知識ブラウザは，知識ベースを静的に解析し，クロスリファレンス情報を表示する機能を持っている(**図3**参照)。知識記述に用いているフレーム名，スロット名などの名称を指定すると，それが定義されている知識ファイルの名前とそのファイル内の位置を表示する。更に，その名称を参照しているフレームやルールがあれば，それらの名前が表示される。知識の修正が必要なときには，知識ブラウザの画面上から直接エディタを起動して即座に修正することができる。

4.2 推論実行中での知識の動的検証(知識デバッガ)

知識デバッガは，意図したとおりに推論が行われているかどうかを検証するためのツールである。知識デバッガは，次に示す三つの機能を持っている。

(1) 推論の実行制御

ブレークポイントでの推論中断と推論再開，1ルールごとのステップ実行などを行うことができる。ルールの実行順序を調べたり，推論の途中でフレーム，ビューノートなどの状態を調査したりすることが可能になる。

(2) 内部状態の表示とその一時的な変更(**図4**参照)

推論過程でのフレームとビューノートの内容を表示し，必要なら変更できる。また，ルールやメタルールの定義や適用可能かどうかを表示することができる。推論過程でのフレームとビューノートの状態が正しいかどうかの確認，及び一時的に事実の修正を行ってからの試行を目的として使用する。

(3) 説明機能

推論結果の導出過程を図示することができる。推論の過程を分析することを目的として使用する。ルール，フレームなどのトレース情報(履歴)を表示したり印刷したりすることができる。

5 専門家自身による知識の修正と追加(知識テーラ)[1)]

エキスパートシステムには，専門家自身の手で知識を追加・修正する環境が要求される。知識テーラは，この要求にこたえる知識獲得ツールである。

エキスパートシステムの知識ベースは，ある程度設計された段階でその構造が決定されている。専門家の知識は，そのなかで，特定のパターンの繰返しになることが多い。専門家自身の手で，

(1) 同じパターンの知識を追加・削除する，あるいは

(2) パターン内のパラメータを修正する，

ことができる必要がある。

ナレッジエンジニアが，エキスパートシステムの知識ベースに現れるパターンを，フォームと呼ばれるメタ知識で定義すると，知識テーラによってカスタマイズドエディタが生成される。カスタマイズドエディタは，フォームに記述されたパターンに従って，上記2種類の修正を可能にする。

カスタマイズドエディタは，画面から入力された内容を知

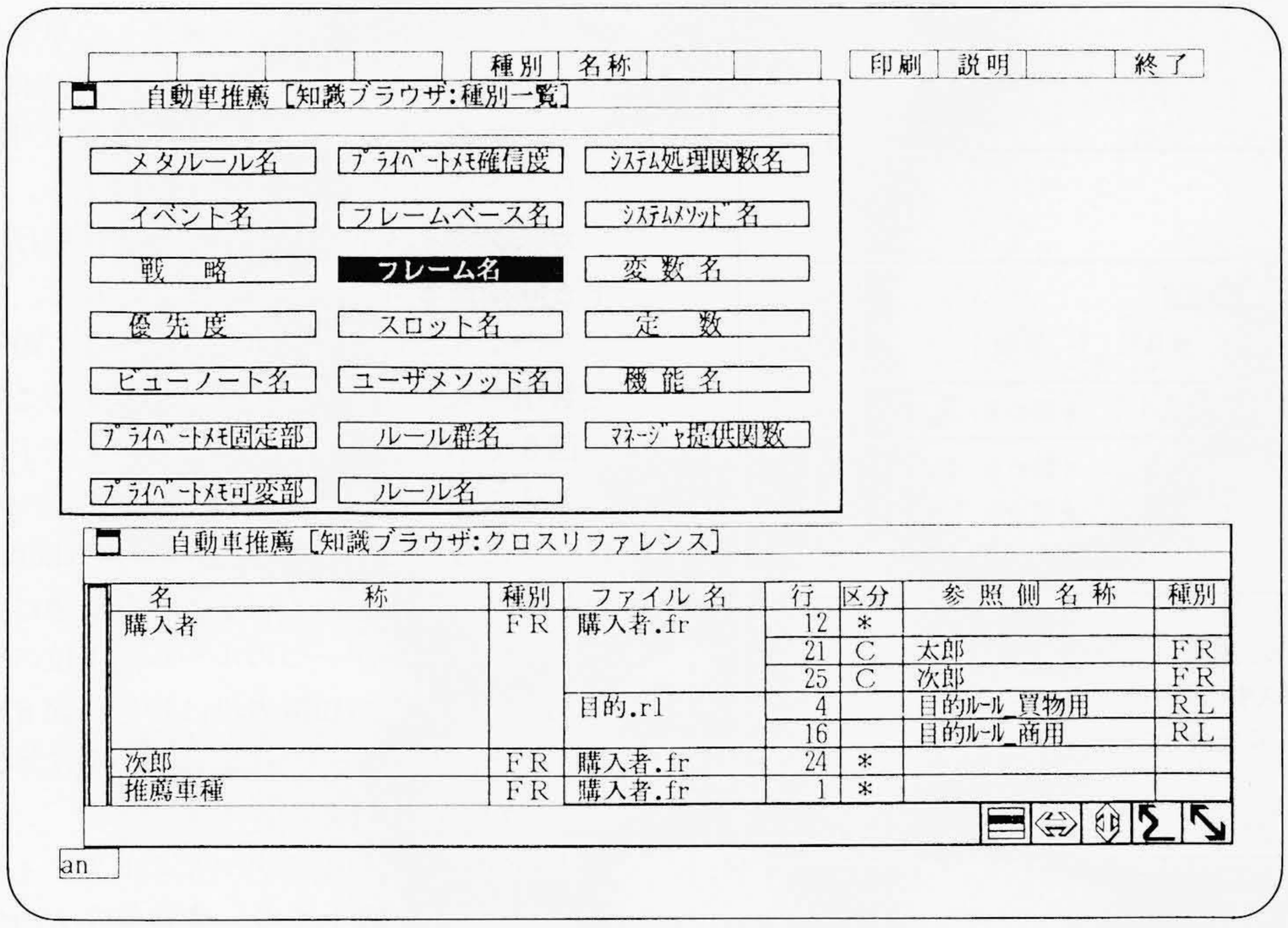


図3　クロスリファレンス情報の表示　　フレーム名のクロスリファレンス情報を取るように指定すると，知識記述に用いた個々のフレーム名について，定義されているファイル名とファイル内の位置，そしてそのフレーム名を参照しているフレームとルールが表示される。

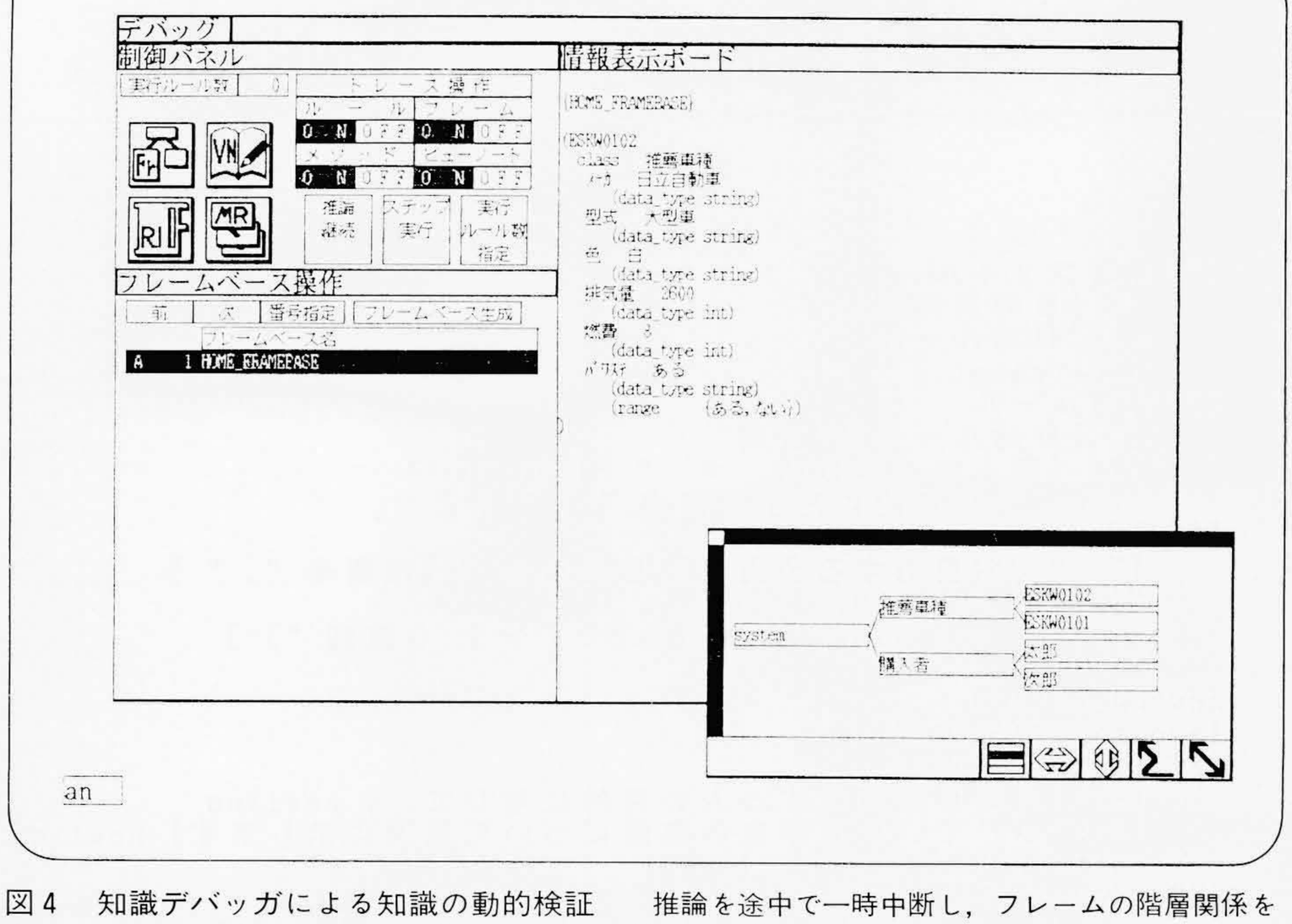


図4　知識デバッガによる知識の動的検証　　推論を途中で一時中断し，フレームの階層関係を表示し，フレーム内のスロットの値を調べたり，値を一時的に変更したりすることができる。

識へ変換するプログラムとして機能する。したがって，画面の仕様と知識のパターンの仕様を与えれば動作する。フォームは，専門家が追加・修正する知識のパターンと，そのパターンを入力するための画面の仕様を記述するための簡易言語である。

専門家が修正するという観点から考えると，知識は修正対象の部分と対象外の部分に分けられる。修正対象の部分を変数と呼び，その値はカスタマイズドエディタの画面から入力

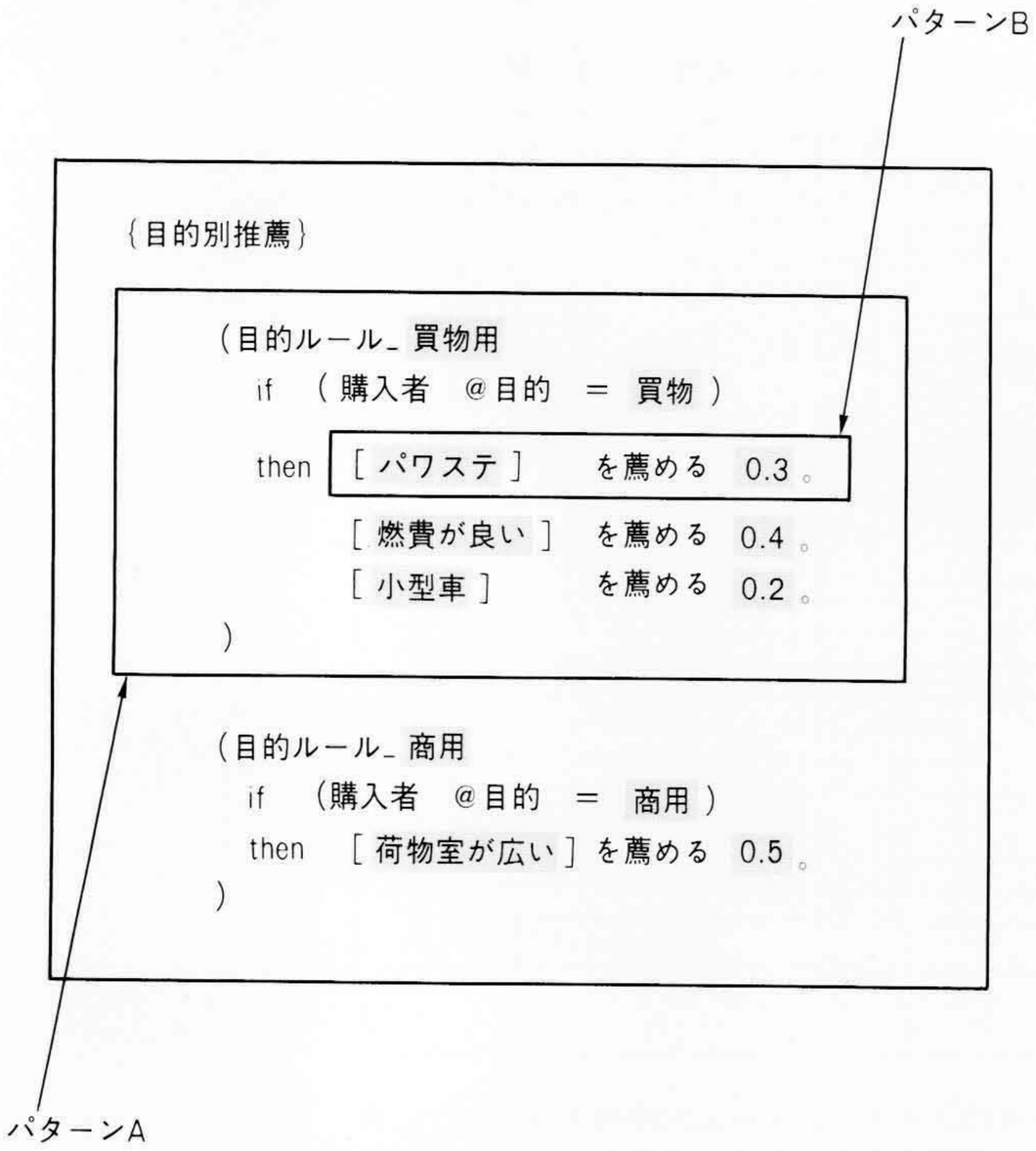


図5　知識表現(ルール)の例　　このルールは上位をパターンA，下位をパターンBとする構造の繰返しである。薄い網目部分が変数で，値を修正する対象となる部分である。

できる。また，同じ構造の繰返しの単位をパターンとして定義することによって知識をパターンごとに追加・削除できる。パターンの一部分が更に繰り返される可能性がある場合は，それを下位のパターンとして定義する。

カスタマイズドエディタの画面仕様は，ソーイングメソッドと呼ばれるもので定義する。これは，変数を専門家に対してどのように質問するかを定義するもので，変数の説明文，変数値の入力位置，表示色を指定することができる。

図5で示すルール用のカスタマイズドエディタを作る場合を考える。このルールを見ると，1ルールが一つの繰返しパターンになっており，そのルールのthen部分がもう一つの繰返しパターンになっていることが簡単に分かる。すなわち，上位のパターンが一つのルール，下位のパターンが一つの実行部である。この知識表現に対し，**図6**に示すフォームを定義すると，知識テーラによって**図7**に示すカスタマイズドエディタが生成される。

専門家は，このカスタマイズドエディタを操作して，パターンを追加・削除したり，変数値を入力することによって知識を修正することができる。

6　ユーザーインタフェースの構築(UIビルダ)

エキスパートシステムも，実用化するためには使いやすいユーザーインタフェースを提供する必要がある。しかし，従

```
[??目的ファイル
    ["{目的別推薦}"
      ??目的ルール
    ] norepeat ]
[??目的ルール
    ["(目的ルール_"?目的"
     if (購入者　@目的　=　"?目的")
     then "??目的に対する配慮")" ]]
[??目的に対する配慮
    ["["?お薦めする装備"] を薦める "?推薦率 "。" ]
    ?推薦率 range[0.0 1.0]
            explain["`推薦率は０．１～１．０範囲`"] ]

sewing_method
layout 目的別推薦
(??目的ファイル
    text(自動車を購入する人の目的に対して、) newline
    text(ついていた方が良い装備について以下に示します) newline
    pattern(??目的ルール))
(??目的ルール
    text("　目的が") variable(?目的) text(であるならば)
    pattern(??目的に対する配慮) text(で薦めます) newline)
(??目的に対する配慮
    text("　") variable(?お薦めする装備) text(を)
    variable(?推薦率) text(ぐらいの重み))
```

図6　フォームの例　　図5でのパターンAが「？？目的ルール」に，パターンBが「？？目的に対する配慮」にそれぞれ対応している。

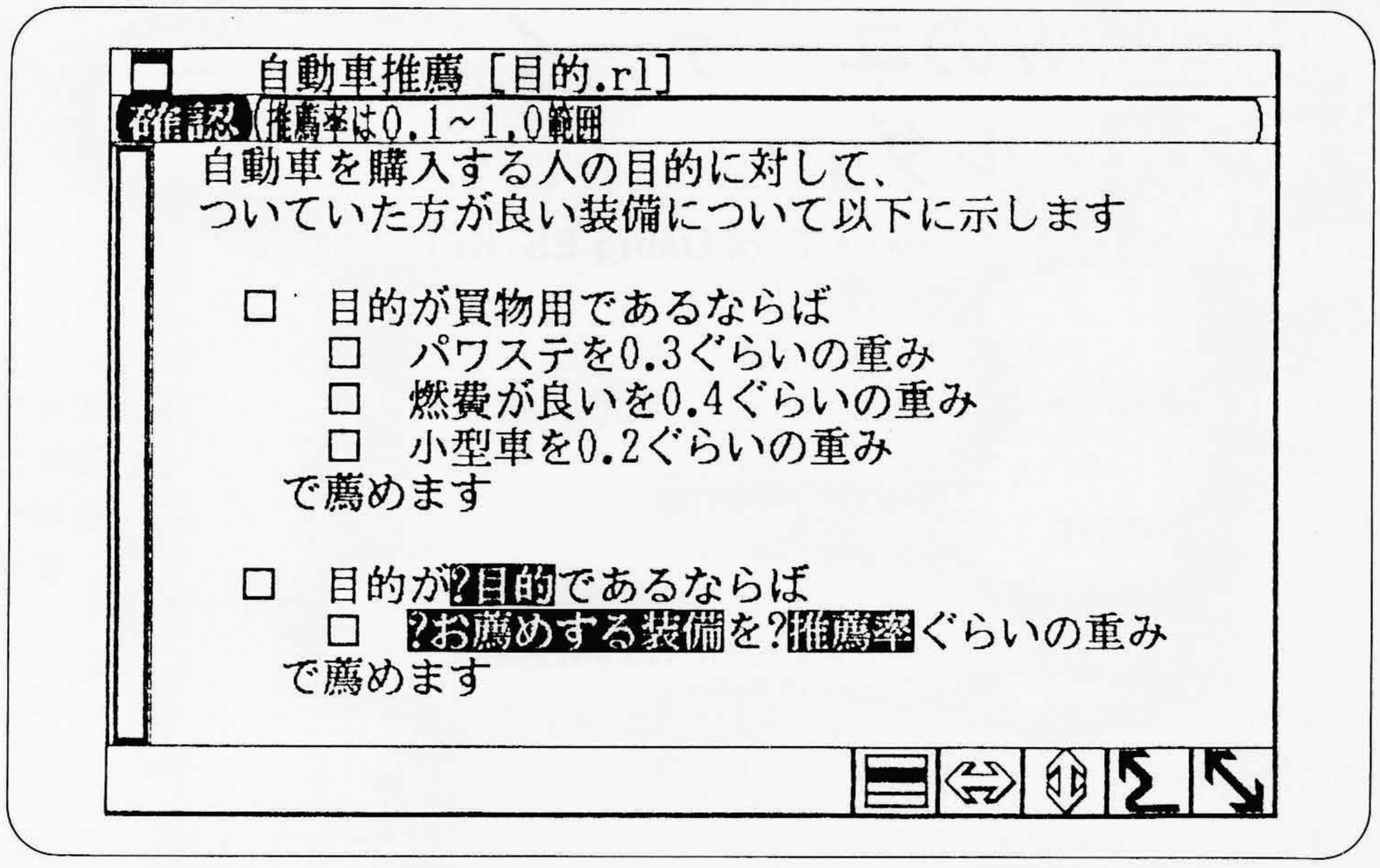


図7　知識テーラによってできたカスタマイズドエディタ　　パターンの追加・削除は，□の部分に対して操作する。変数は白抜きの部分で，ここにキー入力やメニュー選択によって値を設定する。

来，知識処理部分よりもユーザーインタフェースのほうが開発工数が大きいという問題があった。

エキスパートシステムはその特徴の一つとして，システムが容易に変更できることを挙げている。知識が修正されると，ユーザーインタフェースも変更・修正されていくものであり，エキスパートシステムにとってはユーザーインタフェースの保守性が極めて重要である。

この二つの問題を解決するために，ES/KERNEL/WはUIビルダというユーザーインタフェース構築ツールを備えている。詳細は，本特集の別論文(文献[2]参照)に述べている。

7　結　　言

エキスパートシステム構築ツールES/KERNEL/Wの開発環境は，エキスパートシステムの構造モデルに基づいて設計された。エキスパートシステムの構造モデルは，「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伝承するためのメディアである。」という考え方に基づいている。ES/KERNEL/Wは，この構造モデルのコンポーネント一つ一つについて支援機能を提供している。

実用期を迎えたエキスパートシステムは，その応用範囲を急速に拡大している。このため，エキスパートシステムの開発期間を短縮することが必要であり，はん(汎)用のエキスパートシステム構築ツールとしては，その開発環境を充実することが急務であった。ES/KERNEL/Wの開発環境は，この要求にこたえるものである。

参考文献

1)　辻，外：メタ知識定義による知識ベースの保守方式とその適用例，人工知能学会誌，**3**，3(昭63-5)
2)　増石，外：ES/KERNEL/Wのユーザーインタフェース構築ツール「UIビルダ」，日立評論，**70**，11，1100~1104(昭63-11)