

## 法律エキスパートシステム プロトタイプ

吉野 一

明治学院大学 法学部

yoshino@mh.meijigakuin.ac.jp

桜井 成一郎

東京工業大学 大学院社会理工学研究科

古関 義幸

日本電気株式会社 C&C メディア研究所

嶋田 章

日本電気株式会社 第一 C&C システム事業本部

久保田 健一

株式会社アスキング システム開発部

黒田 寿男

システム武蔵株式会社

### 1 はじめに

法律エキスパートシステムの研究を効率よく進めるためには、現行の法律の分析を通して法体系を明らかにしながら、法律知識ベースを記述し、それを説例に対して推論実行を行なう実験を繰り返すことを通して問題点を明らかにしていくことが重要である。

これを行なうためのシステムとして、法律エキスパートプロトタイプシステムを開発してきた。平成6年度には、知識ベースを開発するための環境として、法律知識ベース構築支援システムを開発した。平成7年度には、推論実行を行なう実験システムとして、法律エキスパートシステム・実験システム Ver1.2を開発した。これらの開発を通して、平成8年度までに、国際統一売買法の一部について CPF による知識ベースの開発を行ない、説例に対して正しい法的解釈を行なう推論を行なうことが可能になった。また、平成8年度は、このような法的推論を実現する法律エキスパートシステムは、その推論実行内容を利用者に対してビジュアルに説明す

る機能を持つ必要があるとの認識に基づき、そのインタフェースのあり方を検討する共に、実現方法について検討してきた。

最終年度である本年度は、これらの検討を土台として、昨年度までに開発した法律知識ベース構築支援システムと推論実行システムの充実を図ると共に時間関数記述の拡張などの推論実行機能の拡張を行った。また、国際売買法に基づく知識ベースの拡充を行い、説例に対する法的解釈を行う推論実行を実現した。これらの拡充と同時に、推論説明実行システムを新たに開発し、これを WWW ブラウザを通したネットワークインタフェースにより利用できるようにした。これによって利用者は、推論実行の内容について、各ゴールおよびサブゴールの成功および失敗の原因を容易に把握することができるようになった。また、このシステムはインターネット/イントラネットにおいて世界的に急速に普及しつつある WWW 技術を用いて作られているため、ネットワークを通して、プラットフォームに独立のインターネット上の様々な端

末(ブラウザ)から利用できるようになっている。

本項では、まずはじめに本年度に開発した推論説明実行システムについて述べ、その後知識ベース作成支援・推論実行システムについて述べる。

### 2 推論説明実行システム

#### 2.1 システム構成

本システムは、HTTP サーバ、推論ゲートウェイ (CGI プログラム)、推論エンジンからなるサーバ部と、マン・マシンインタフェースを司るブラウザ部から成る (図1)。

このうち HTTP サーバと推論ゲートウェイは同一の機械上に置く必要があるが、推論エンジンは、ソケット通信を介して推論ゲートウェイとのやりとりを行うように設計されているため、ネットワーク上の任意の機械の上に実装することができる。もちろん WWW ブラウザについても同様である。

今回の実装に於いては、サーバ部を1台のパソコン上に実装した。OS に FreeBSD-2.2.2R を、http サーバに apache-1.2b10 を採用した。推論ゲートウェイ (CGI プログラム) と推論エンジンの記述言語として Prolog を選び、処理系には SICStus-3.5 を利用した<sup>1</sup>。

WWW ブラウザとしては主に Netscape 3.0 Gold (X11) を中心に試験を行ったが、ユーザが好みに応じたブラウザを自由に選択できるように、ハイパーテキストの記述には HTML のみを使用している。このため、WindowsOS を搭載したパソコンからも、インターネット/イントラネットを介して利用が可能になっている。

#### 2.2 推論エンジン

推論エンジンは、CPF を直接実行する Prolog メタインタプリタとして実装されている。初期値として、CPF ルールファイルとゴールファイル、ソケットのポート番号が与えられ、起動後はデーモンとして常駐する。ソケット通信を介したネットワーク上のプロセスからの要求に対してメタインタプリタが呼び出され、結果を返すことができる。

<sup>1</sup> CGI は一部 perl でも書かれている。perl は perl-5.002 を利用した。

推論エンジンを CGI プログラム (推論ゲートウェイ<sup>2</sup>) から分け、そのインタフェースをソケット通信にしたことで、プログラムの独立性が高まっている。また、SICStus Prolog で記述されているので、プログラムソースは特定のプラットフォームに依存しない。

以下、推論エンジンと推論ゲートウェイとの対話プロトコルについて記す<sup>3</sup>。

##### 2.2.1 推論エンジンと推論ゲートウェイとの対話プロトコル

推論エンジンと推論ゲートウェイとの対話のシンタクスは CPF の項 (Prolog の項の拡張である) に従う。

推論エンジンは起動されると、最初に socket を bind する。その後以下を繰り返す (図2)。

1. listen 状態に入り、推論ゲートウェイからの要求を待つ。
2. 要求が来たら accept を返し、入力 stream から要求を読む。
3. 要求の内容に従って、結果を出力 stream に書き出す。
4. 入力 stream に end of stream を検出すると stream を close し、再び listen 状態に入る。

推論エンジンは、以下のような要求を受け付ける。

- halt  
エンジン停止。
- set\_of\_goal  
ゴール ID の一覧 (リスト) を返す。  
set\_of\_goal(Set)
- goal(ID)  
ID で指示されたゴールを返す。  
goal(ID,X,Y)

<sup>2</sup> 第2.3節参照

<sup>3</sup> 推論エンジンの CPF メタインタプリタとしての機能は過去に何度か解説されてきているので、本稿では割愛する

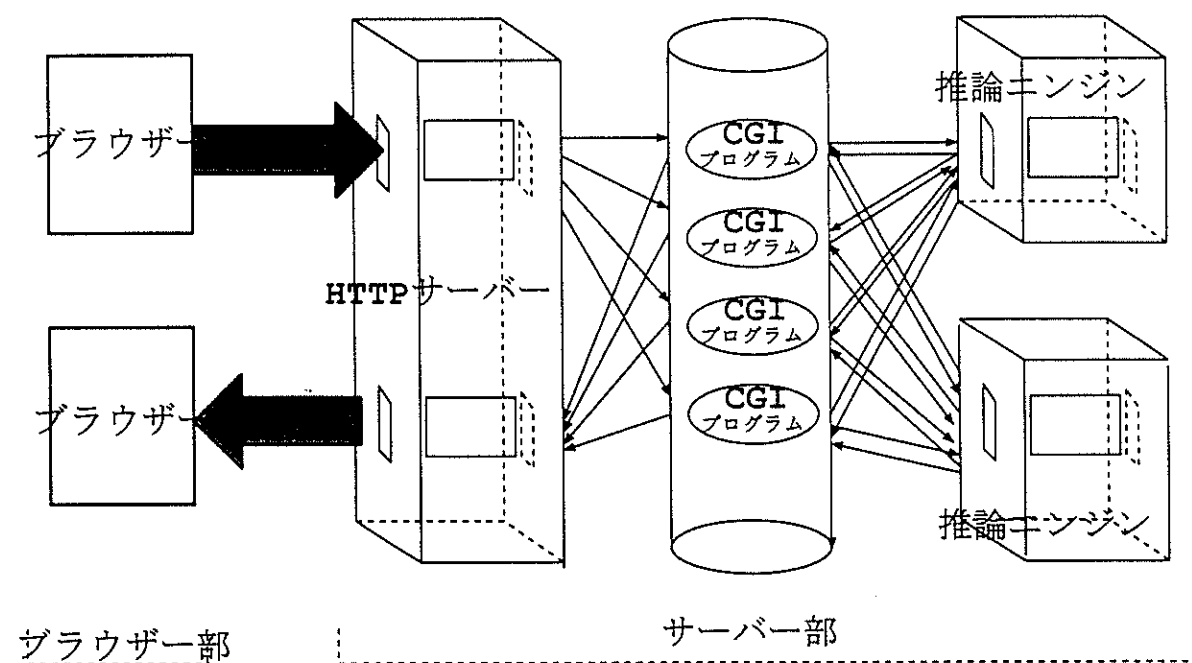


図 1: システム構成

- rule(ID)  
ID で指示されたルールを返す。  
rule(ID,X,Y,Z)
- fact(ID)  
ID で指示されたファクトを返す。  
fact(ID,X,Y)
- explain(ID,Level)  
ID で指示されたゴールを推論する。返すべき推論結果 (推論木) の深さを Level で指示する。  
proof(Proof)
- explain2(Goal,Level,Time)  
Goal で与えられたゴールを推論する。時間の制約リストが Time で与えられる。返すべき推論結果 (推論木) の深さを Level で指示する。  
proof(Proof)

### 2.3 推論ゲートウェイ

推論ゲートウェイは CGI プログラム群として実装されており、HTTP サーバと推論エンジンとの間をとりもつ。推論エンジンとの通信

は第 2.2 節で述べた。本節では (推論ゲートウェイを成す) 各 CGI プログラムを取り上げ、それぞれの機能について解説する。

#### 2.3.1 初期画面

推論エンジン (ポート番号) とそれを動作させるホストを指定する。表示するゴール、ルールおよびファクトの表示モード (CPF 表示か日本語表示かの切り替え) もここで指定する (図 3)。

#### 2.3.2 ゴール一覧表示 CGI

初期画面 (図 3) から Submit されると、ゴール ID のリストを推論エンジンから得、HTML として出力する (図 4)。

ゴール ID が選択され、Submit ボタンが押されると、推論結果表示 CGI が起動されることによりこのゴールが実行され、その結果が説明画面として表示される (図 6、図 7)。

#### 2.3.3 帯図表示 CGI

前節で述べたようにゴール一覧リストから実行すべきゴールを選択するインタフェースでは、

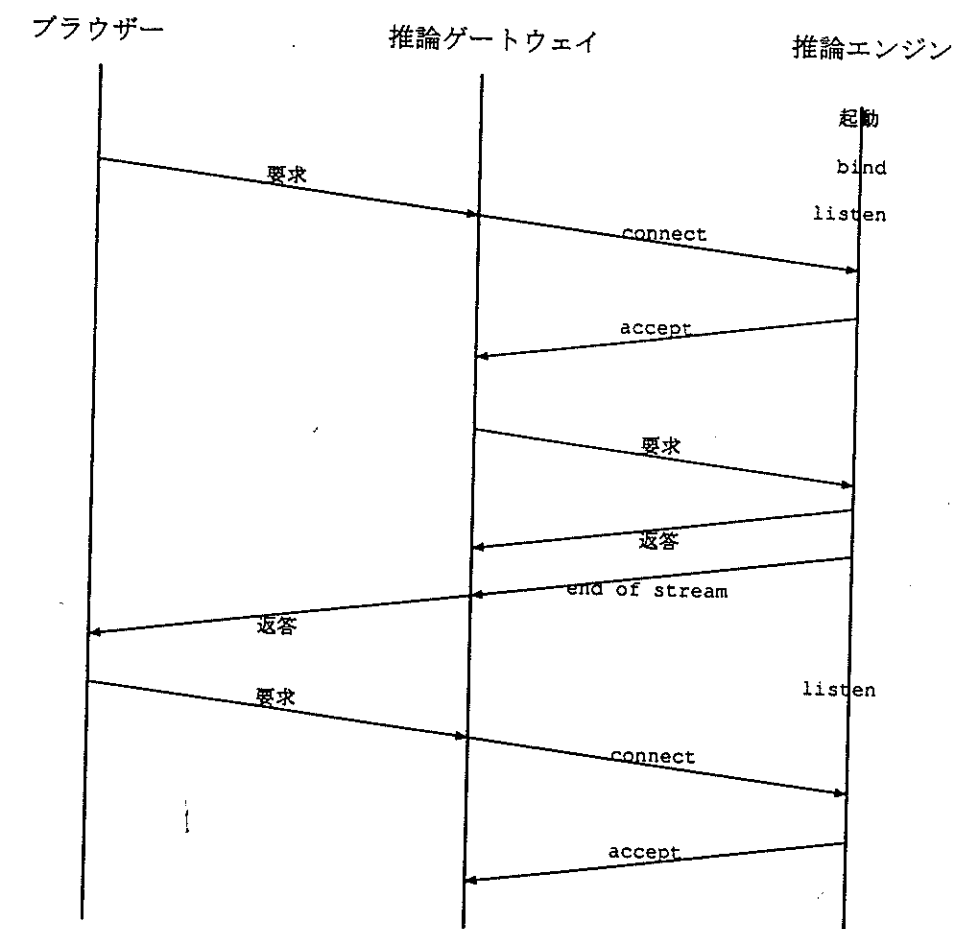


図 2: 対話プロトコル図

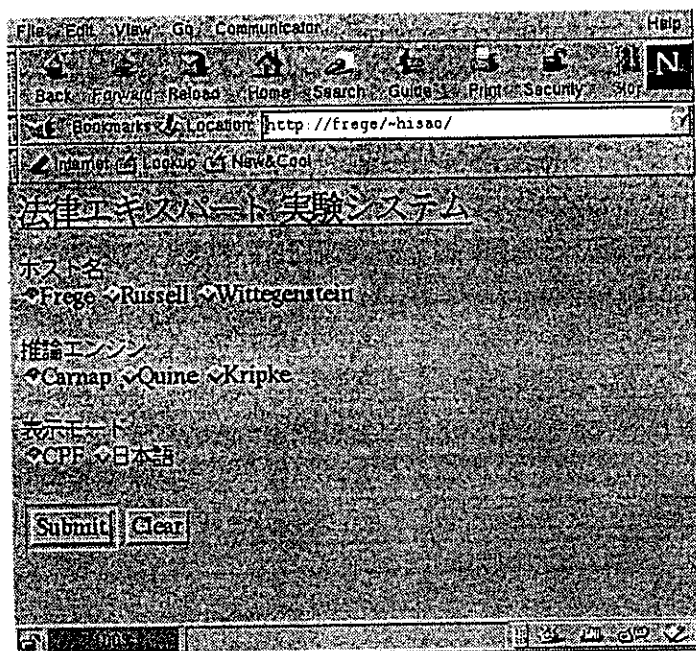


図 3: 初期画面

法律関係に基づくゴール間の関係が理解できない。そこで、法的関係の時間的推移をビジュアルに表現し、そこからゴール呼び出しを行えるようなインタフェースを設ける方式を考案した。時間的推移を表現する方法として、法的権利および義務について、その効力の発生から消滅までを時間の帯図として表現する方式をとる。これを、HTMLの表形式記述を用いて実現することにした。図5はその帯図の一例である。帯図は各時点でのアンカーを選択することで上で記述した説明機能呼び出しが可能となっている。

#### 2.3.4 推論結果表示 CGI

上記のウィンドウからゴールが選択されると、そのゴールをパラメータとして、説明推論エンジンに対して推論要求が出される。説明推論エンジンは、そのゴールを使って説明推論を行い、その結果をHTML形式に変換して出力する。表示出力は、推論結果表示説明部(図6)と推論結果表示ゴール部(推論エンジンにかけられる前のゴールを表示する画面、図7)に分かれて、同じ画面に上下段に出力される。

推論結果表示説明部では、推論エンジンの推論結果の説明が表示される(図6)。ここでは、

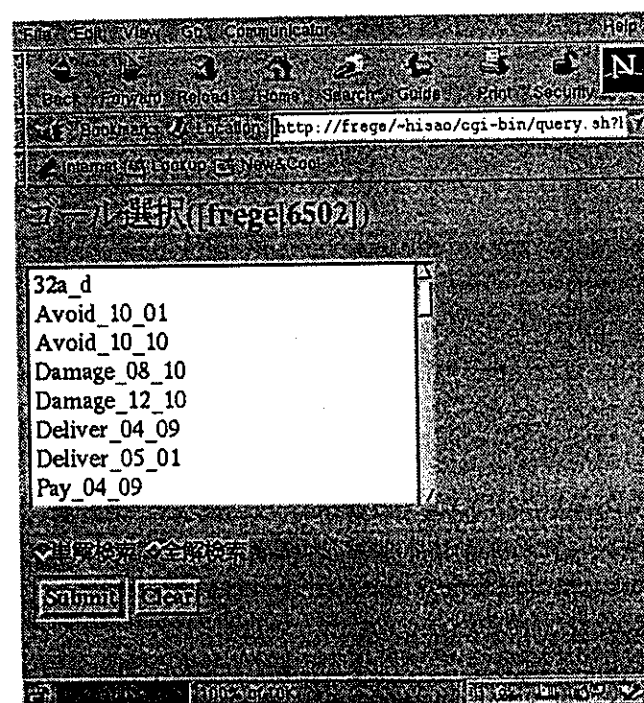


図 4: ゴール一覧

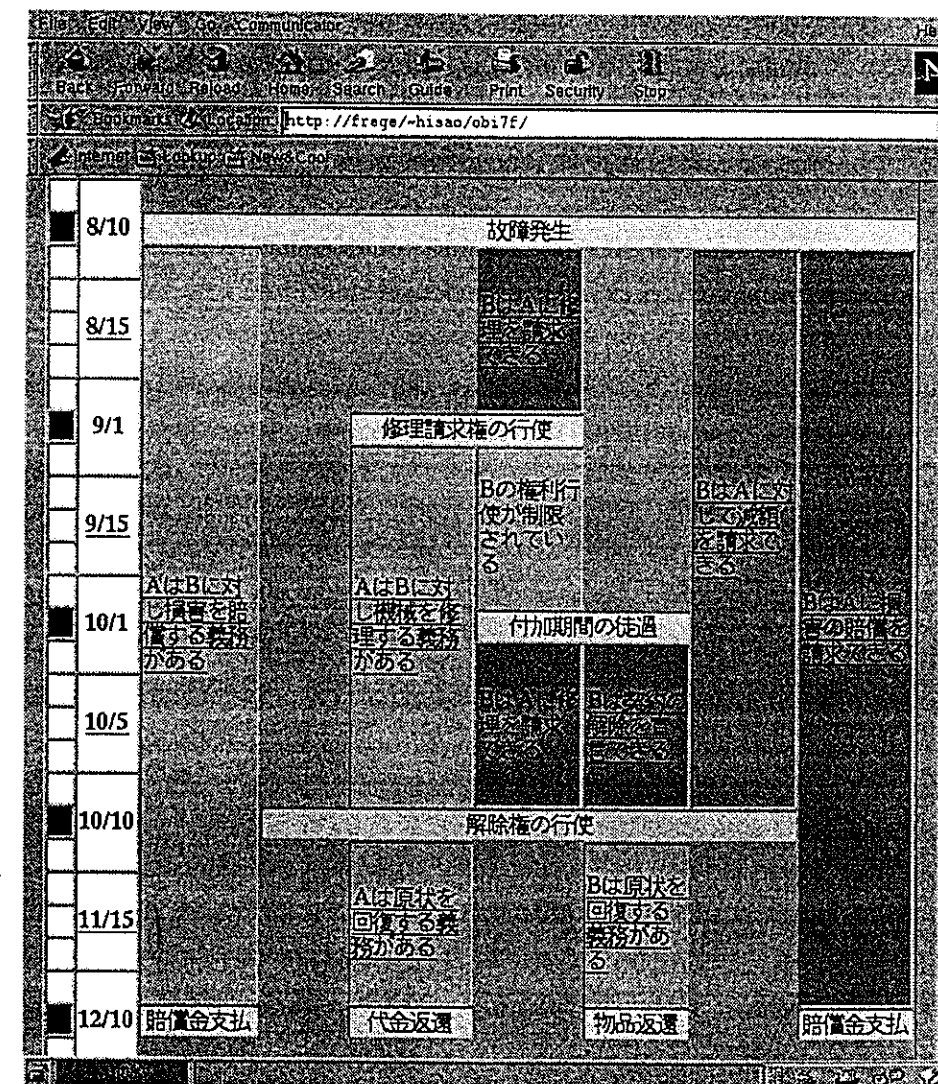


図 5: 帯図表示画面

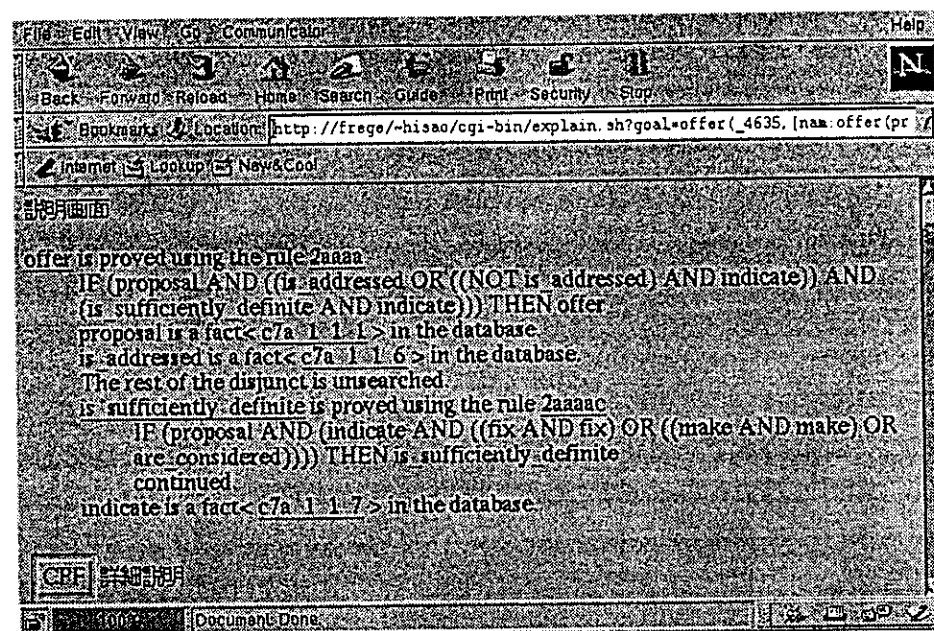


図 6: 推論結果表示説明部

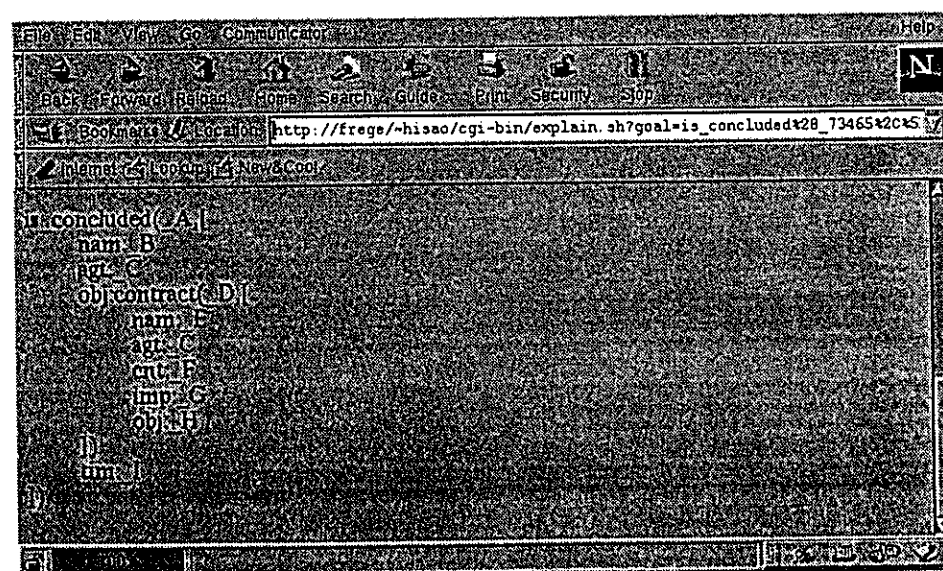


図 7: 推論結果表示ゴール部

ルールが成功した場合に加えて失敗した場合にも説明が選られるようになっている。

まずゴールの成功 / 失敗が書かれ、そのゴールにどのルールが適用されたかがルール ID によって表示される。このゴール部分は、アンカーとなっており、マウスクリックにより、証明されたゴールが別ウィンドウに表示される。また、ルール ID もアンカーとなっており、クリックすることにより、そのルールの内容が CPF 形式で別ウィンドウ表示される。

適用ルールの内容の要約が表示された後、続いてそのルールがなぜ成功したか、あるいは失敗したかが、サブゴール毎の実行結果として表示される。

サブゴールがファクトにマッチして成功した場合には、ファクト ID が表示される。このファクト ID をクリックすることにより、ファクト表示 CGI によって図 8 のような画面が表示される。これらにより、実際にどのようなファクトにマッチしてサブゴールが成功したかがわかる。

サブゴールがルールにマッチして成功した場合は、その述語名を示したアンカーが表示されると共に、適用したルールのルール ID とそのルールの概要が表示される。述語名アンカーをクリックすることにより、再帰的に推論結果表示 CGI が起動され、下位レベルの説明が得られる。サブゴール表示のネストの深さはデフォルトで 1 になっているので、一段下のサブゴールの結果のみが表示されるが、必要ならばパラメーターで深さを指定することもできる。また、ルール ID をクリックすることにより、ルール表示 CGI によって、図 9 のようなルール表示画面が表示される。これにより実際にどのようなルールが適用されたがわかる。

またゴールが成功した場合には、『CPF』詳細説明ボタンを押すことで変数の束縛状態を含めて、証明された CPF の詳細が別ウィンドウ (図 10) に表示される。

## 2.4 拡張性

以下今後の拡張性について検討する。

### 2.4.1 自然言語インターフェース

本システムのユーザインターフェース部は、HTML と CGI プログラムで記述してあるため

に推論エンジンと独立に変更あるいは新規開発が行なえる。各サイトの好みや技術に応じてユーザインターフェース部を作り込めば、例えば自然言語によるエンジンとの会話が可能になる。

自然言語インターフェースについては、昨年度開発を行ったテンプレート方式による変換プログラムを流用して、CPF を変換テーブルに基づいて簡易自然言語に変換して表示するプログラムを現在試作中である。図 11 に CPF で書かれたルールを簡易日本語に変換して表示した例を示す。

### 2.4.2 CPF デバッガとしての機能充実

今回実装した説明機能は一種のステップとして働く。したがってこれに、ブレークポイントの設定 / 解除や実行機能等を加えることで、CPF のソースレベルデバッガとしての機能を持たせることができる。

### 2.4.3 法的推論サーバと分散推論

現在、推論エンジンに対してのクライアントは推論ゲートウェイ (CGI プログラム) しか考えられていないが、インターフェースがソケット通信で切っているために原理的にはネットワーク透過な任意のプロセスがこれを利用できる。インターフェースを適切に設計することができれば法的推論サーバとしてワールドワイドに利用することが可能となる。

また、推論エンジンはネットワークの任意のノードに位置させることができるため、これを、解かれるべき 1 つのゴールに対してネットワークの複数ノードに置くことで、分散並行な推論を行うシステムに拡張することも可能である。

## 3 知識ベース作成支援・推論実行システム

法律エキスパートシステムの中核は法的推論システムである。それは、法律家が行う法的推論の実際にできるだけ近似な推論をコンピュータ上に実現するものである。これを実現していくためには、現行の法律の分析を通して法対系を明らかにしながら、法律知識ベースを記述し、それを説例に対して推論実行を行なう実験を繰り返すことを通して問題点を明らかにしていくと

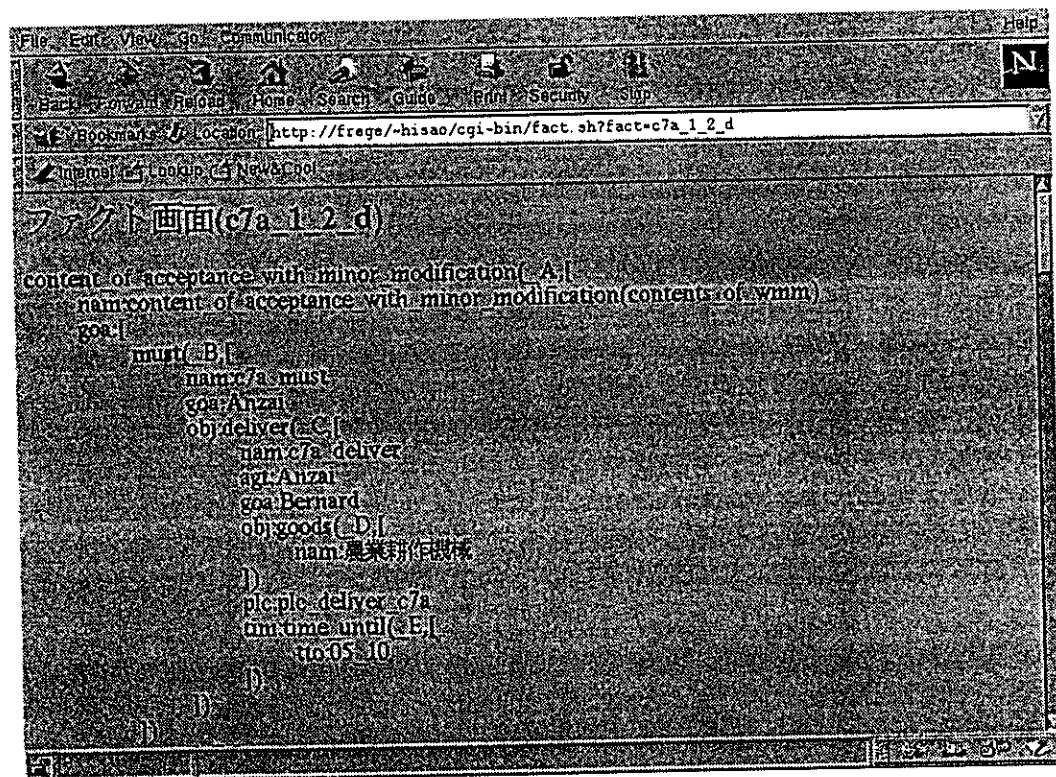


図 8: ファクト表示画面

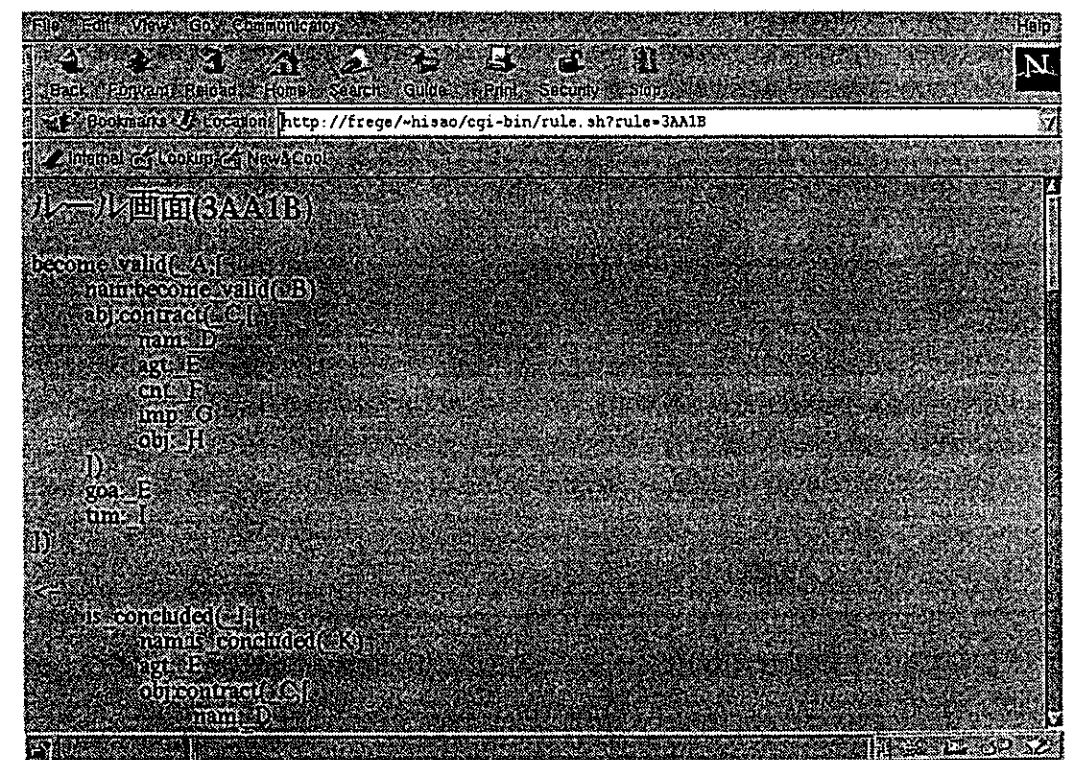


図 9: ルール表示画面



ともに、知識ベースと推論方式を改良していくことが重要である。これを目的として、知識ベース作成支援および推論実行システムが開発された。

### 3.1 システム構成

本システムは、知識ベース作成支援と推論実行システムとから構成される。知識ベースエディタ (図 12~図 13) を中心とする知識ベース作成支援システムと、CPF 表記された知識ベースを解釈して法的推論を実行する推論実行システムとから成る。

なお、知識ベース作成支援システムは Tcl/Tk が動作する環境で、推論システムは SICStus Prolog が動作する環境で、利用可能である。

図 12の左側のダイアログ (Legal Expert System Supervisor) は、このシステム全体のメインメニューである。メインメニューの「Edit Data」ボタンをクリックすると、左側のダイアログ (Edit Data) が現れる。Edit Data ダイアログの「Rule Editor」ボタンをクリックすると図 13の知識エディターメイン画面が表示される。

#### ソフトウェア構成

|         |                     |
|---------|---------------------|
| OS      | UNIX                |
| メインメニュー | Tcl/Tk              |
| 知識エディタ  | Tcl/Tk (一部機能に C 言語) |
| 推論エンジン  | SICStus Prolog      |

### 3.2 推論実行システム

推論実行システムは、CPF で記述された知識ベースを利用し、同じく CPF 表記された説例ファクトファイルに対して、推論実行を行なう。また、デバッグおよび推論の理解を目的として、トレース機能、スパイ機能、ビジュアルトレース機能を持つ。これらの表示は、日本語テンプレート機能を利用することにより、簡易自然言語文として表示することができるため、CPF ないしは Prolog でそのまま表示する場合に比べてはるかに可読性の高い形式で表示ができる。また、述語の概念階層を用いて知識ベースをマクロ展開することにより、簡潔な表現で知識ベースの記述が可能になっている。また、

各種推論パラメータの設定、実行の制御などを、GUI を用いた推論用ダイアログウィンドウにより行なえるので、使いやすくなっている。

#### 3.2.1 推論エンジン

推論エンジンは、CPF を直接実行する Prolog メタインタプリタとして実装されている。SICStus Prolog で記述されているのでポータビリティがよい。ここでは、推論ダイアログウィンドウ、簡易自然言語表示機能、階層展開、ファクトファイルフラット化について説明する。

#### 3.2.2 推論ダイアログウィンドウ

図 14は、知識エディターの「Debugger」メニュー内の「Files for Inference」ボタンをクリックすることにより表示される推論実行ダイアログである。この GUI を通して、以下のような機能が制御できる。

- ルール、ゴール、ファクト、Taxonomy などの各ファイルをあらかじめ推論用に指定しておく。
- 使用する推論エンジンのタイプを選択する。但し、現在のところ一種類のエンジンのみ。
- 推論トレース実行時のトレースの深さを設定できる。数値により任意のレベルが指定できる。
- ルール中の全コンセプトを予めサーチしておき、スパイポイント設定ダイアログで選択して、スパイポイントを設定する。設定は、あらかじめルールファイルを検索して作られたメニューを通して行なえるので、マウスにより選択するだけで、タイプをせずに入力できる。
- 「Quick run」ボタンを押すと、ゴールファイルで指定されているゴールが一覧表示されるのでそれを選択すると、そのゴールについての推論が実行される (図 15)。これも、あらかじめルールファイルを検索して作られたメニューを通して行なえるので、マウスにより選択するだけで、タイプをせ

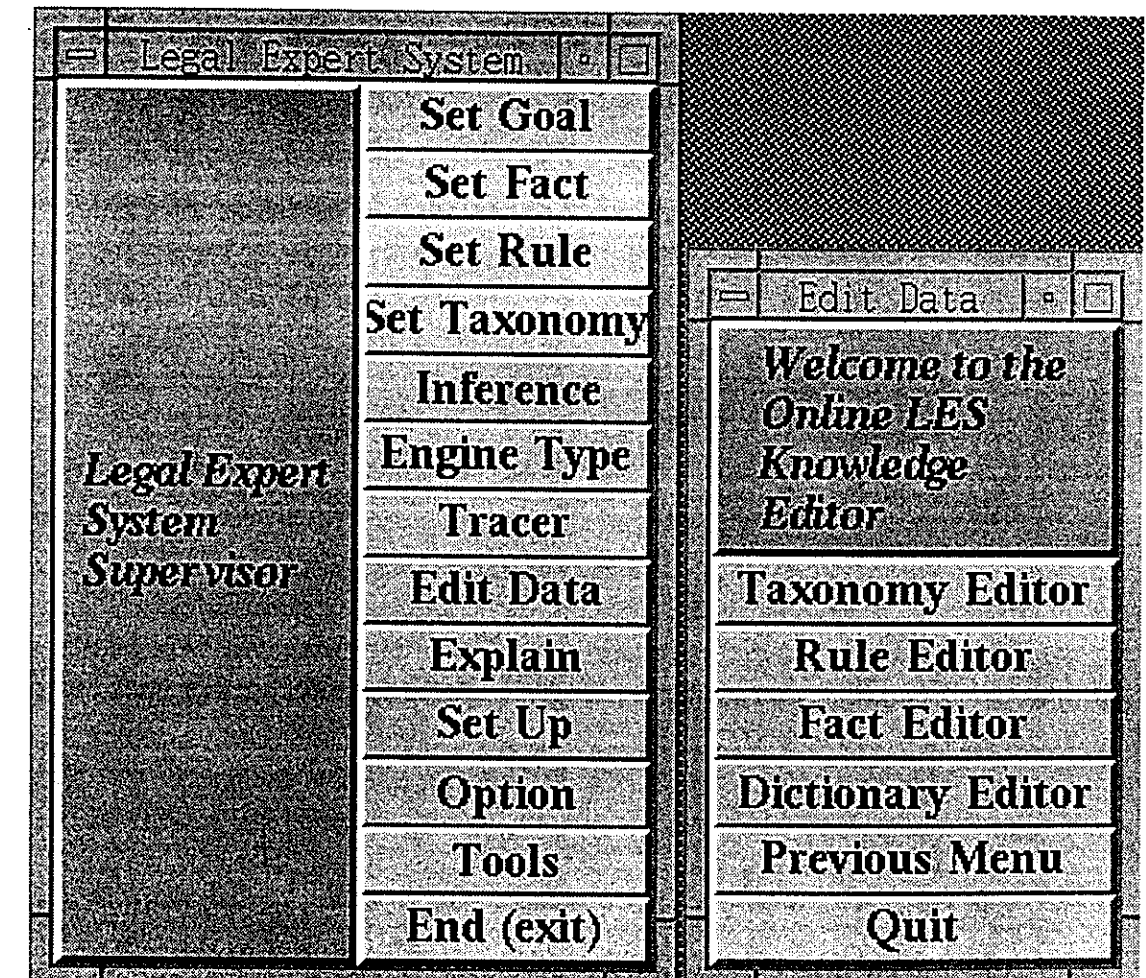


図 12: システムメニュー画面

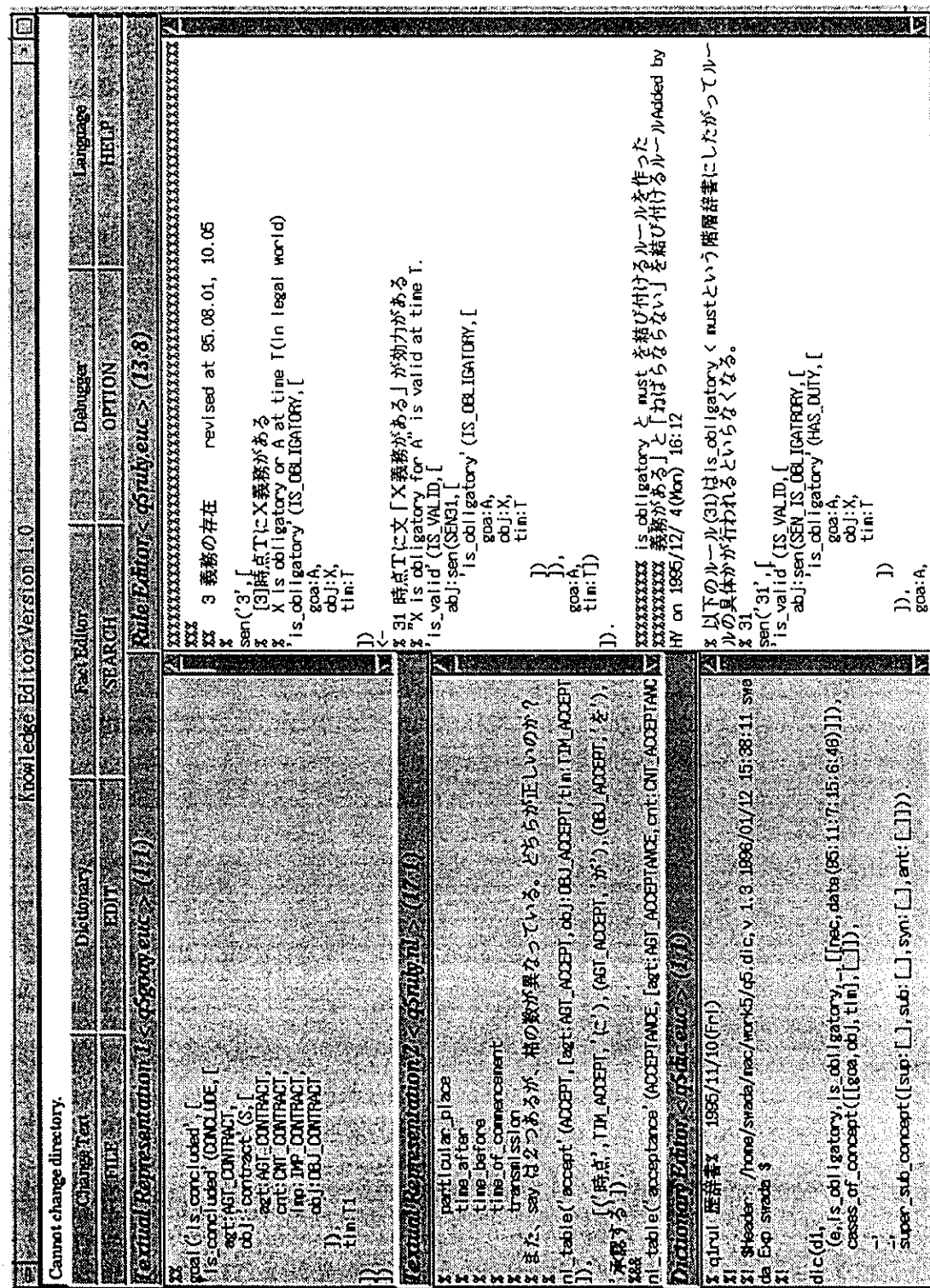


図 13: 知識エディタ画面

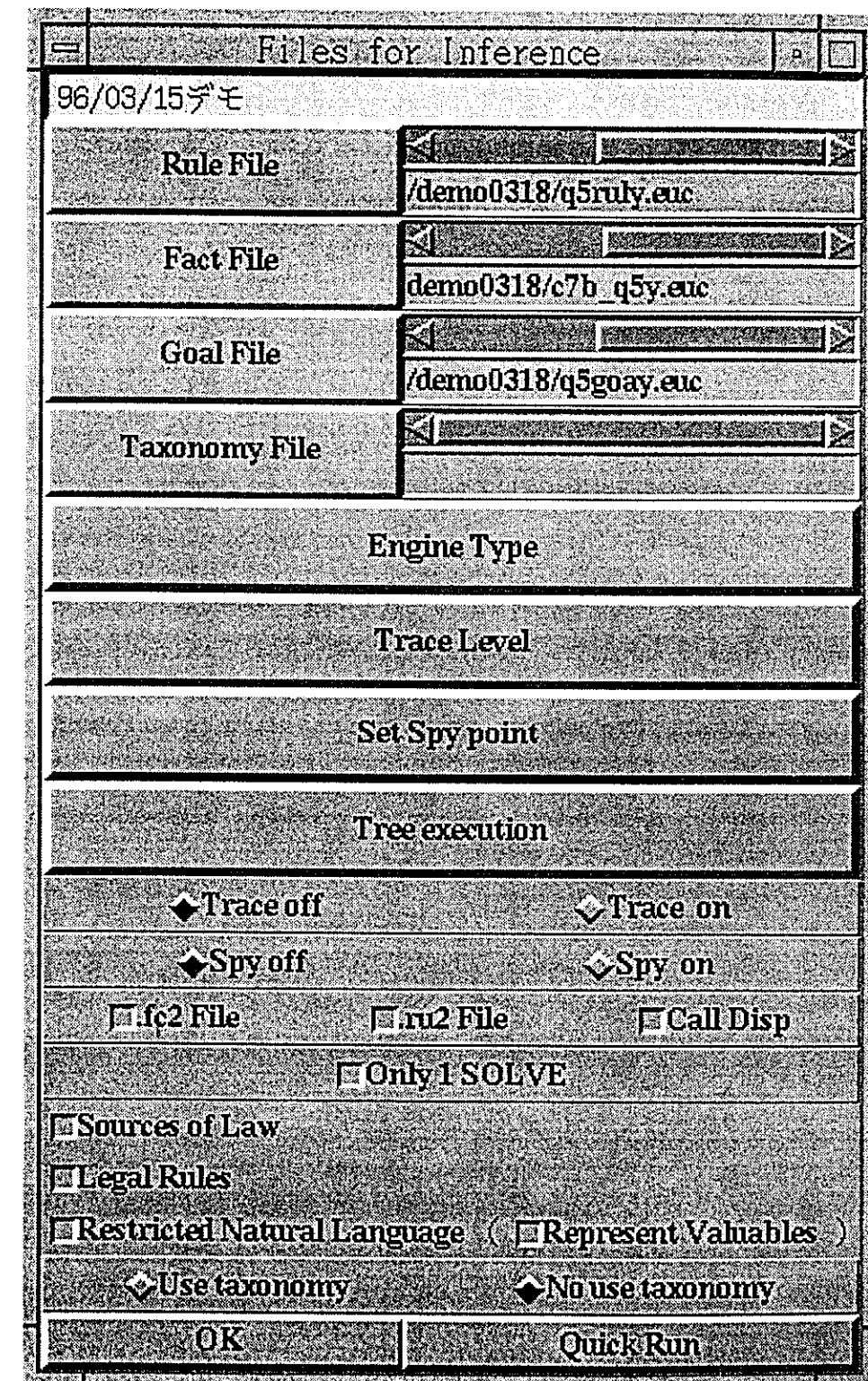


図 14: 推論実行ダイアログ

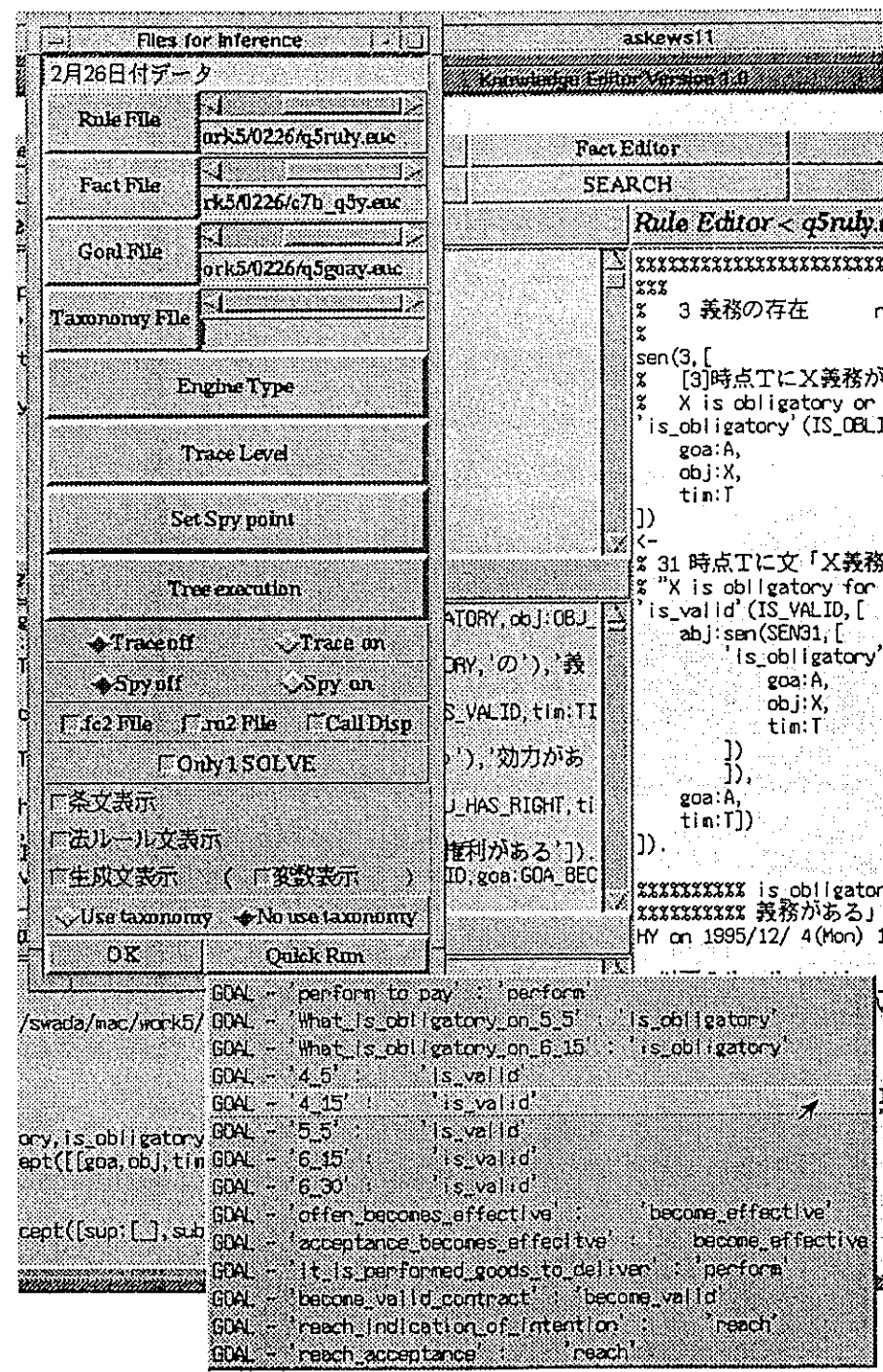


図 15: ゴール実行

ずに入力できる。

- ビジュアルトレーサを起動して、推論実行のトレースを、ツリー表示機能を使ってビジュアルに表示する。(図 16)  
図中の箱は推論実行をトレースする形で次々に描画される。

- 実行制御スイッチにより、以下のような実行時制御を行なえる。

- トレースの ON/OFF
- スパイの ON/OFF
- フラット化ファクトの参照の ON/OFF
- マクロ化ルールの参照の ON/OFF
- 解答一時停止スイッチ (答えが 1 つ出るとに一時停止) の ON/OFF
- 自然言語表示の ON/OFF

### 3.2.3 簡易自然言語表示機能

CPF で記述されたルール、ゴール、ファクトを、自然言語テンプレートを利用して簡易自然言語に変換して表示する機能を開発した。これにより、CPF そのものでの表示に比べて、格段に可読性が向上した。これを推論システムに組み込み、推論エンジン実行時のトレースの表示、ゴールの表示などに利用している。また、説明のためのビジュアルトレーサの表示にも利用する予定である。例えば、下記の例題のように CPF が入れ子になっている場合でも簡易自然言語による表示が可能である。

<元のルール>

```
'is_valid'(IS_VALID,[
  abj:sen(SEN31,[
    'is_obligatory'(IS_OBLIGATORY,[
      goa:A,
      obj:X,
      tim:T])
    ]),
  goa:A,
  tim:T
]).
```

<自然言語テーブル>

```
nl_table('is_obligatory'(IS_OBLIGATORY,
  [goa:GOA,obj:OBJ,tim:TIM]),
  ['「', (OBJ), '義務がある」']).
```

```
nl_table('is_valid'(IS_VALID,
  [abj:ABJ,goa:GOA,tim:TIM]),
  [( '時点',TIM,'に'),(ABJ,'が'), '効力がある']).
```

<ルールから生成される自然言語>

時点 T に「X 義務がある」が効力がある

ここで、自然言語テーブルで定義される表示形式内で変数を扱うことができることに注意されたい。変換実行時に変数のままである場合は変数名表示を行ない、インスタンス化されている場合は、その値を表示する。なお、変数がインスタンス化されていない場合にはその部分を表示しないモードも選択できるようにしている。これにより不要な変数を表示しないようにできる。

図 17は、ルール毎に自然言語を表示する機能の実行画面である。コンセプト名をマウスで選択し、「Debugger」メニュー内の「Natural Language (for test)」コマンドを選択すると、そのルールについて登録されている自然言語が図 6 のように表示される。

図 18は、「生成文表示」ボタンを ON にして Quick Run のゴールを選択したあとの画面である。「条文表示」「法ルール文表示」「生成文表示」のいずれかのボタンが ON になっている時にゴールを選択すると、選択されたゴールについての自然言語を表示してから推論実行に移る。

図 19は推論画面 (自然言語表示モード) である。左側の推論実行ウィンドウ上に成功したタームが現れた時に、それに対応する自然言語を右側のウィンドウに表示する。条文ウィンドウは、ルールに対応する条文を表示する。法ルール文ウィンドウは、ルールに付随しているコメントを表示する。生成文ウィンドウは、ルールの自然言語テーブルをもとにしてシステムが生成した文を表示する。

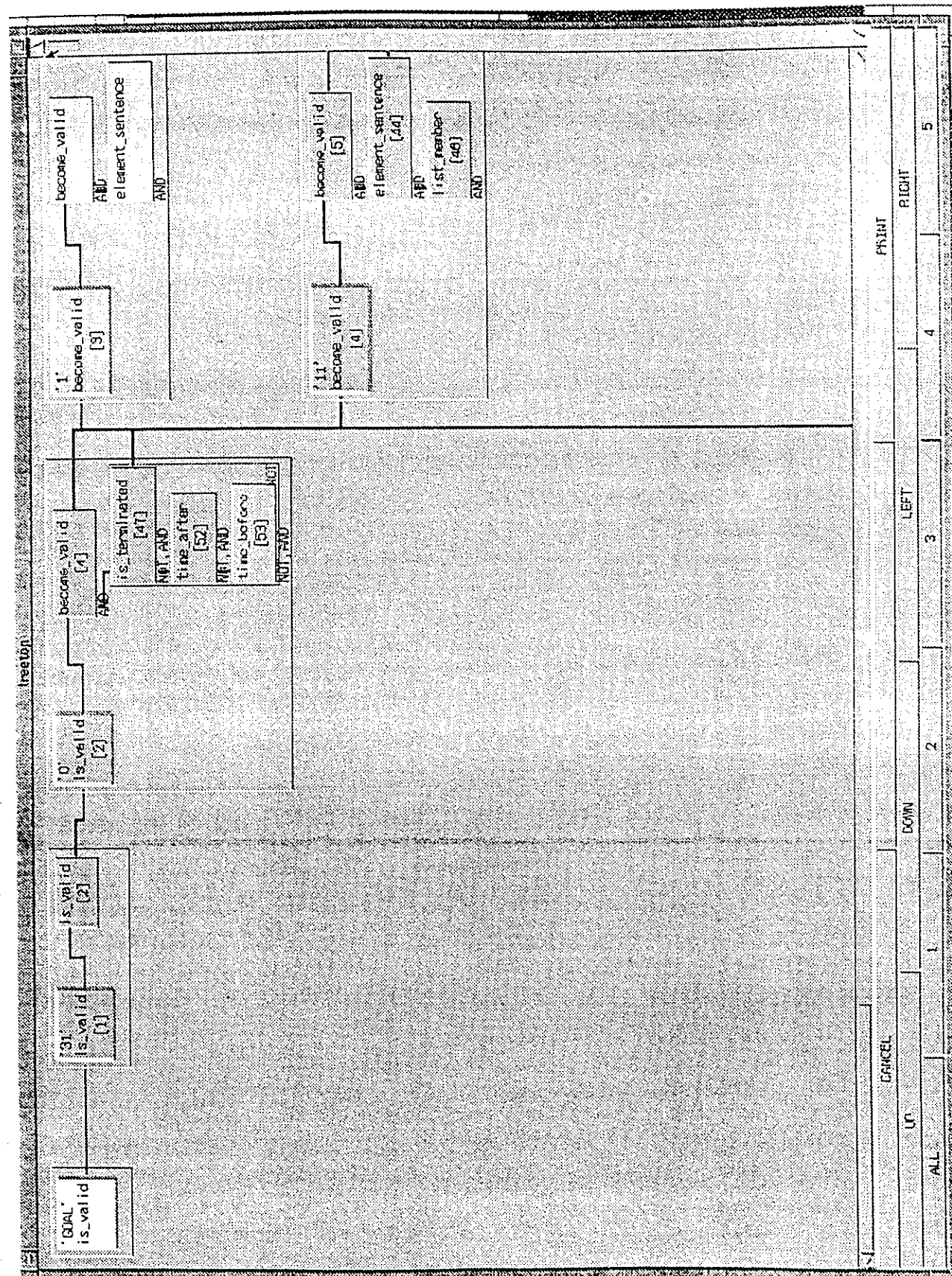


図 16: ビジュアルトレーサ画面

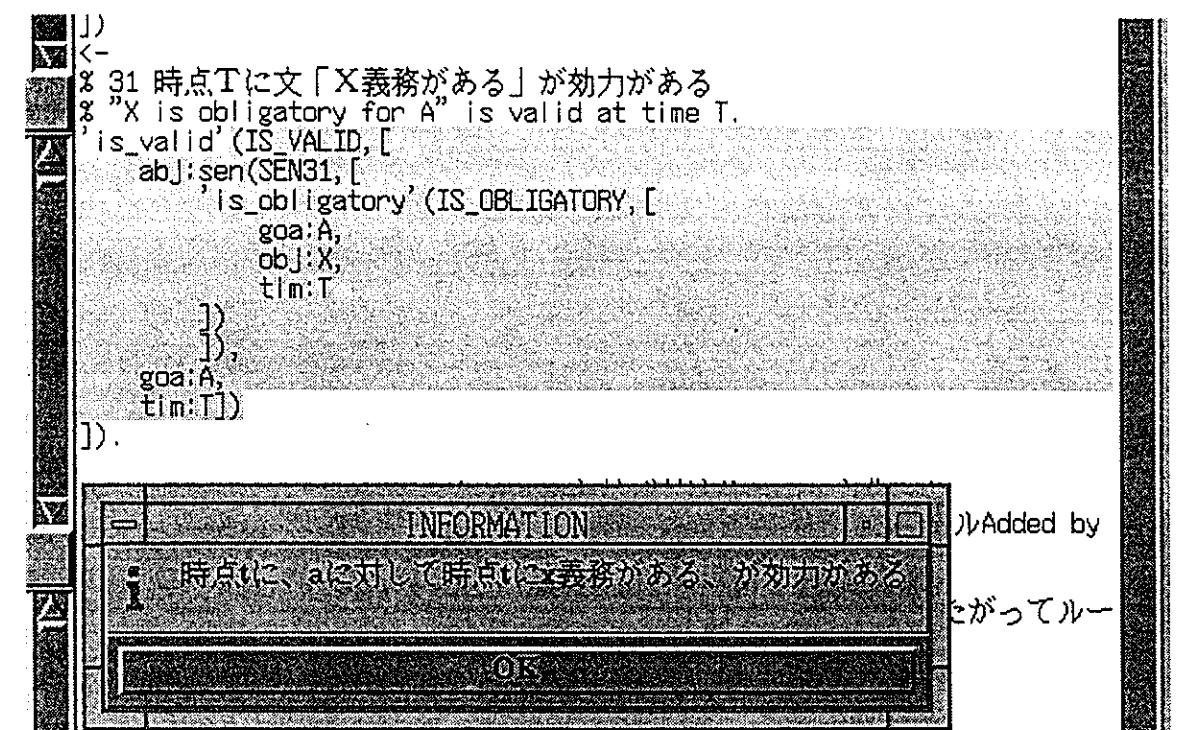


図 17: ルールの自然言語表示

### 3.3 知識ベース作成支援システム

知識ベース作成支援システムは知識ベースエディタとして機能するだけでなく、以下のような機能拡張も持っている。

#### 3.3.1 辞書からのルールタームの生成

マウスで辞書ファイルの一部を選択し、この機能を実行することにより、CPFのタームの枠組を自動作成し、ルールエディタに挿入する。この機能により、ルールの作成が迅速化されるだけでなく、タイプミスなどによる誤りが防止できるようになった。

#### 実行例

<辞書例>

```
dic(d1,
  (e,is_obligatory,
    is_obligatory,
```

```
  ),
  [[nec,date(95:11:7:15:6:46)]],
  cases_of_concept([[goa,obj,tim],[_]])が現れて、シンタックスチェックを行う。
```

```
-,
-,
super_sub_concept([
  sup:[_],sub:[_],syn:[_],ant:[_]))).
```

↓ 変換

```
'is_obligatory'(IS_OBLIGATORY,[
  goa:GOA_IS_OBLIGATORY,
  obj:OBJ_IS_OBLIGATORY,
  tim:TIM_IS_OBLIGATORY
])
```

#### 3.3.2 ルール/ファクトチェック

知識ベースエディタ上に読み込まれているルールまたはファクトの文法チェックを行う機能がある。

図 20は、ルールチェックが終了した時の画面である。知識エディタの「Debugger」メニュー内の「Rule/Fact check」コマンドを起動すると図 20のようなチェッカーウィンドウ

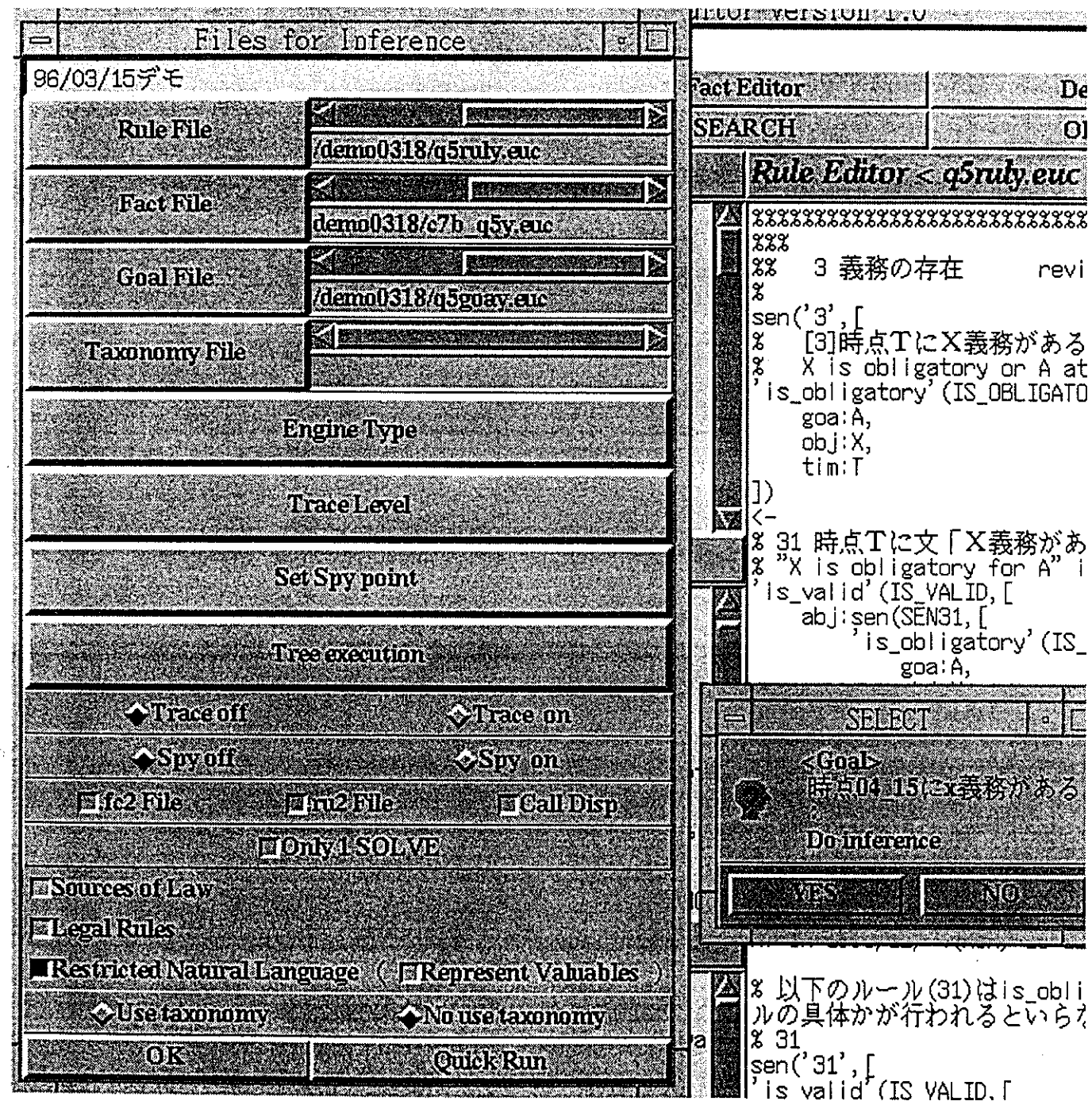


図 18: ゴールの自然言語表示

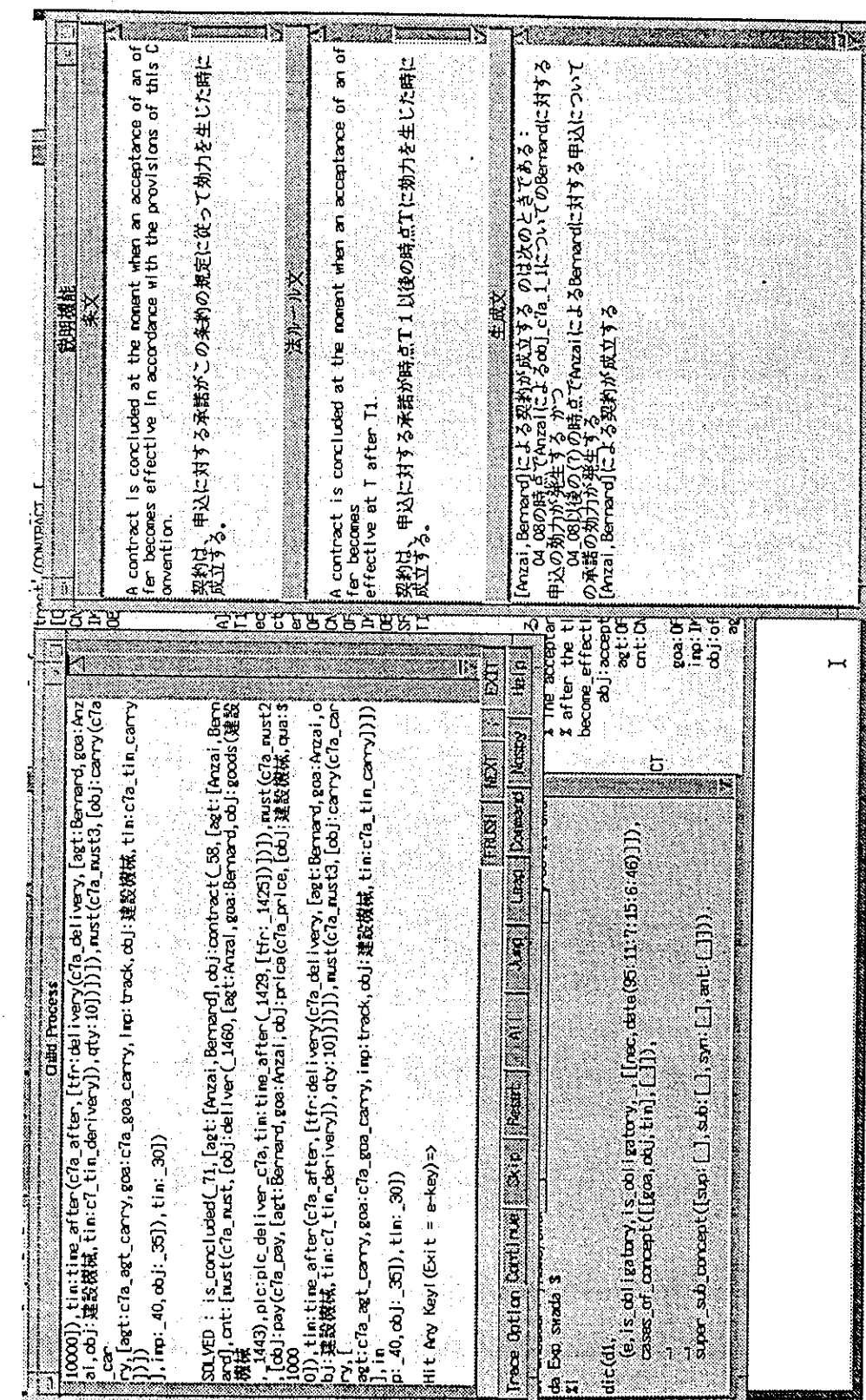


図 19: 推論結果の自然言語表示

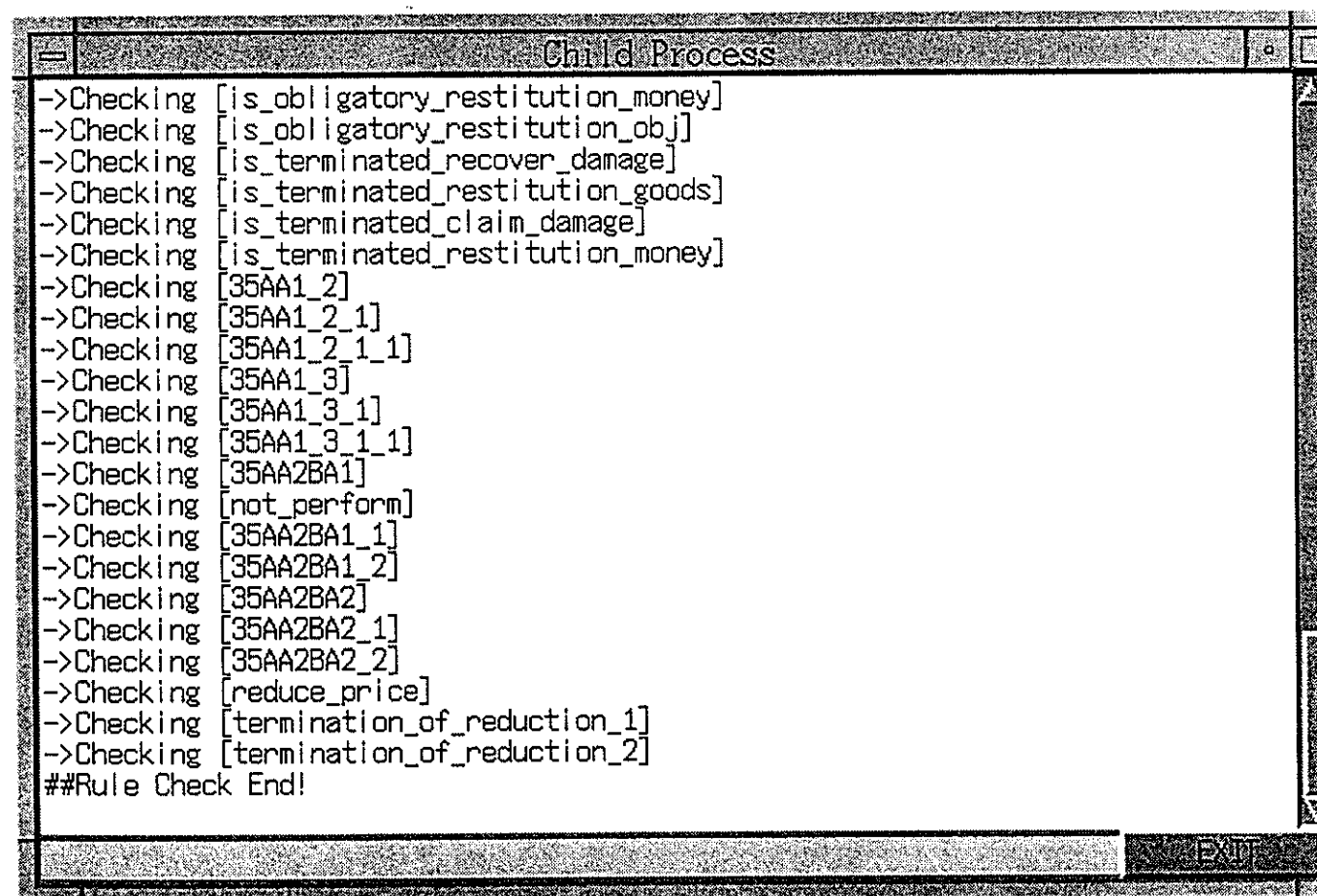


図 20: 文法チェック機能

### 3.3.3 辞書チェック

知識ベースエディタ上に読み込まれている辞書の文法チェックを行う機能がある。

## 4 おわりに

本項では、法律エキスパートシステムプロトタイプシステムについて、その知識ベース作成支援システム、推論実行システム、および推論説明実行システムについて述べた。知識ベース作成支援システムを用いることにより、CPFに基づいた知識ベースを効率よく開発できるようになった。推論実行システムにより、知識ベースと説例に基づいた法的推論が可能になった。また、本年度に開発した説明機能により、WWWブラウザからネットワークを通して、推論実行の内容について容易に把握することができるようになった。本システムの利用実験を通して、本アプローチの有効性を確認できたとともに、いくつかの将来の研究課題が明らかになった。