

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、その後に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・マシンインター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¹。

WWW ブラウザとしては主に FreeBSD 上の Netscape Communicator 4.04 を中心に試験を行った。極力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 のみを使用しており、Windows OS 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基本的には可能になるように努力した。しかし、日本語環境との相性などの問題により、必ずしもすべての環境での動作は保証されていない。

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。

推論エンジンを CGI プログラム (推論ゲートウェイ²) から分け、そのインター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³。

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。

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)

²第 2.3 節参照

³推論エンジンの CPF 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

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

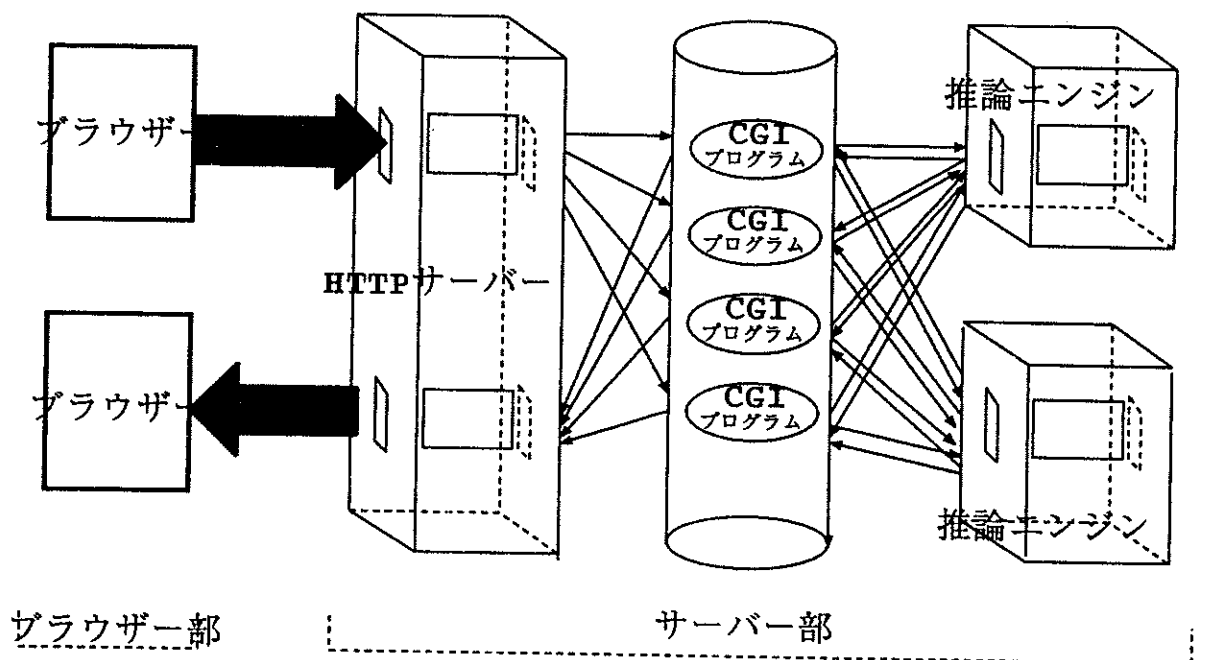


図 1: 推論説明実行システムのシステム構成

- goal(ID)
ID で指示されたゴールを返す。
goal(ID,X,Y)
- 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。推論ゲートウェイを成す各 CGI プログラムは独立して起動する必要があるため、推論のモード、表示言語の選択などのパラメータは、CGI の引数および Web ブラウザの Cookie 機能を用いて毎回推論システムに指示する必要がある。

2.4 画面表示

本節では、画面の表示例を示しながら説明実行システムの動作を説明する。

2.4.1 初期画面

法律エキスパートシステムのホームページからシステムの実行画面をたどって行くと、(図 2) のような画面が表示される。この画面では、適用する法律名、適用する学説、適用する事例を選択する。法律名の選択は読み込むべきルールファイルを選択する。適用する学説の選択は同様にルールファイルを選択するインタフェースであるが、現時点はこの選択機能は未実装で

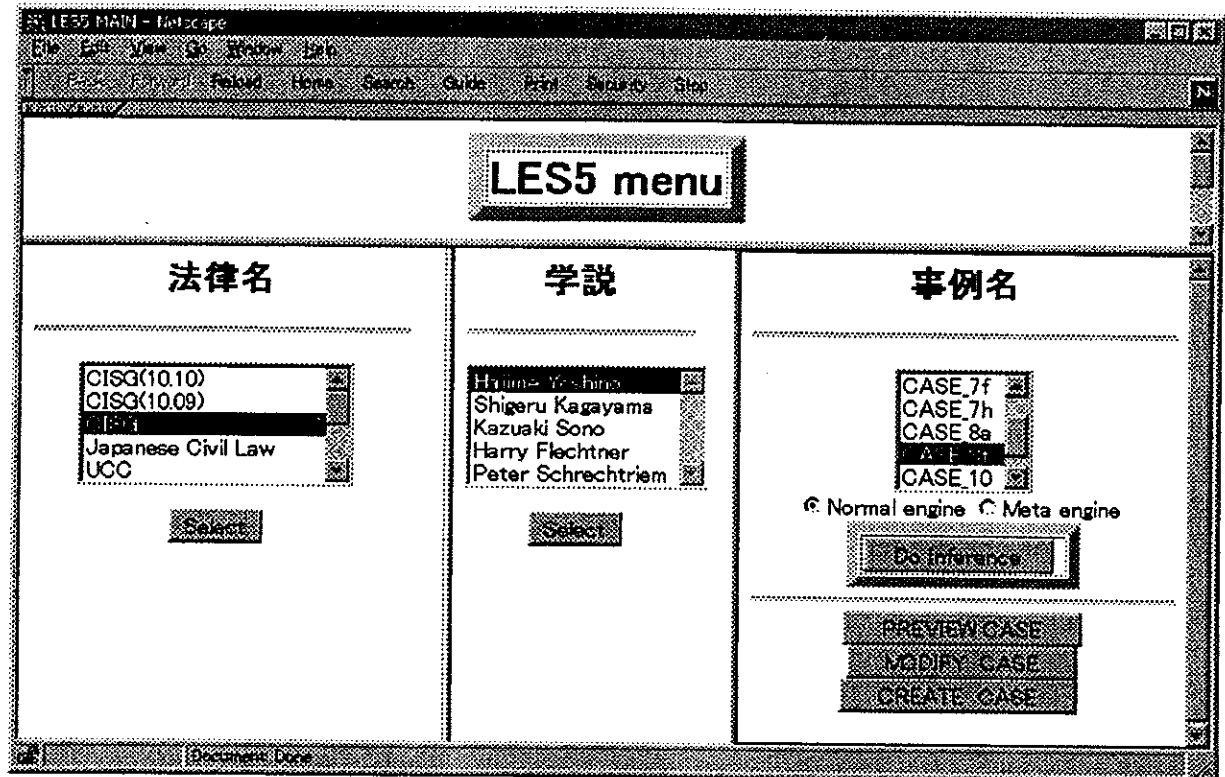


図 2: 初期画面

ある。事例名は読み込むべきファクトファイルの選択を行なう。「Do Inference」ボタンを押すことにより、サーバが起動され実行が始まる。このとき実行される推論エンジンとしては、通常の推論エンジンとメタ推論機能を持ったエンジンの2種類が用意されており、これはラジオボタンで選択できる。また、イントラネット内での利用を想定して、選択された事例ファイルの内容を表示 (PREVIEW CASE) したり (図 3)、内容を修正 (MODIFY CASE) したり、新しく作成 (CREATE CASE) したりすることも可能なインタフェースになっている。但し、このようなサーバ上のファイルを利用者が自由に操作できるようなインタフェースは、今後インターネット上を通して外部からの利用を許可する際にはファイル破壊の危険などに対して考慮をする必要がある。

2.4.2 自然言語変換システムの選択

サーバが実行されると自然言語変換システムの選択画面が表示される。(図 4)。これは、推

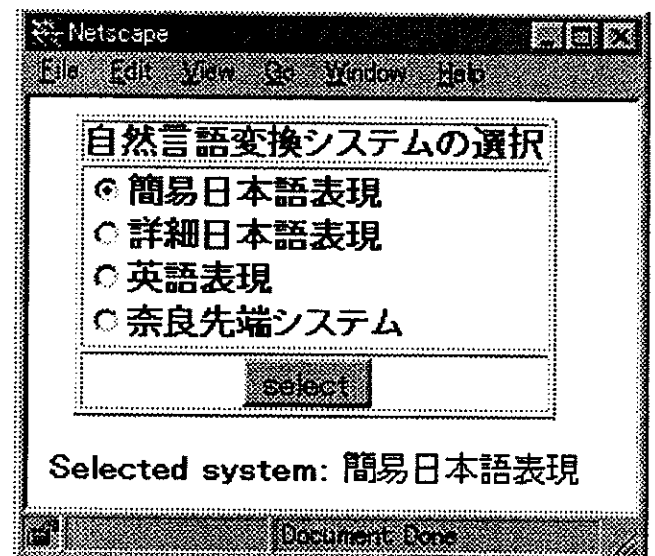


図 4: 自然言語変換システムの選択

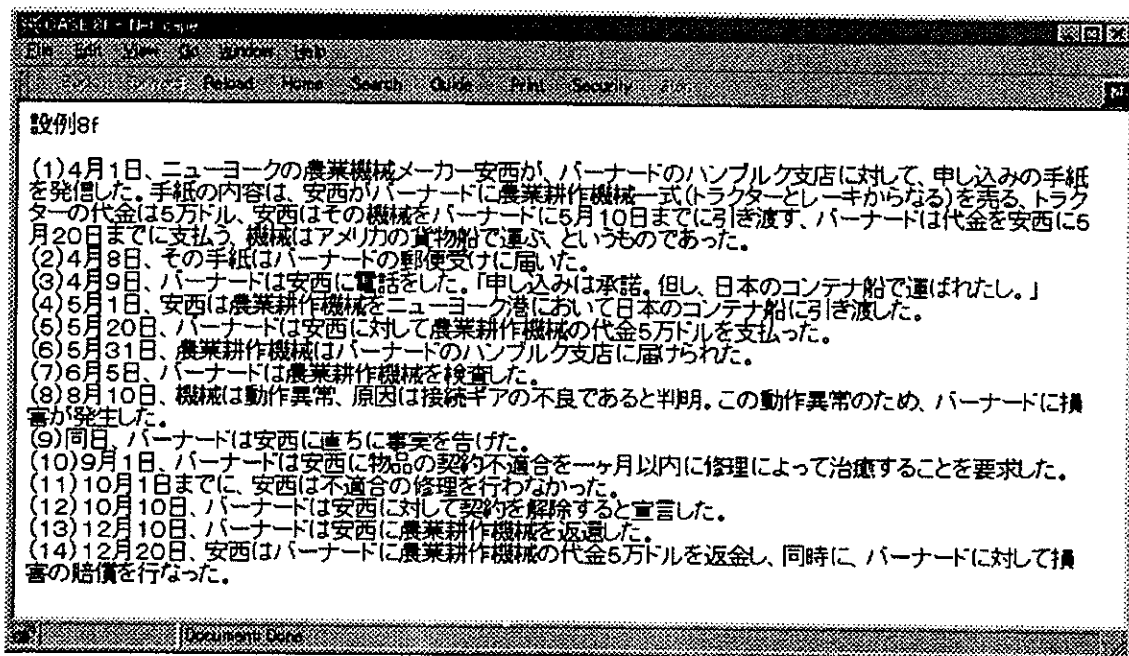


図 3: 事例

論実行結果を表示する際に、CPF 表現を自然言語に変換する方式を選択するものである。簡易日本語表現と詳細日本語表現は変換テーブルを用いたテンプレート方式による変換プログラムを起動する。簡易と詳細の区別は、変換テーブルをそれぞれ別に用意することで実現している。英語表現も同様に英語への変換テーブルを指定することで実現している。また、本格的な自然言語処理による自然言語変換システムを利用できるようにインタフェースを用意しているので、将来 CPF からの自然言語変換システムが実現された場合にもこのインタフェースからの指定で利用できるようになっている。現在、この自然言語変換システムは未実装である。「Select」ボタンが押されると、指定された変換システムの ID が Cookie に登録される。指定された以降、推論が実行されるたびに、指定された変換システムを通った結果が表示される。従って、サブゴールの実行毎に変換システムを変更できる。

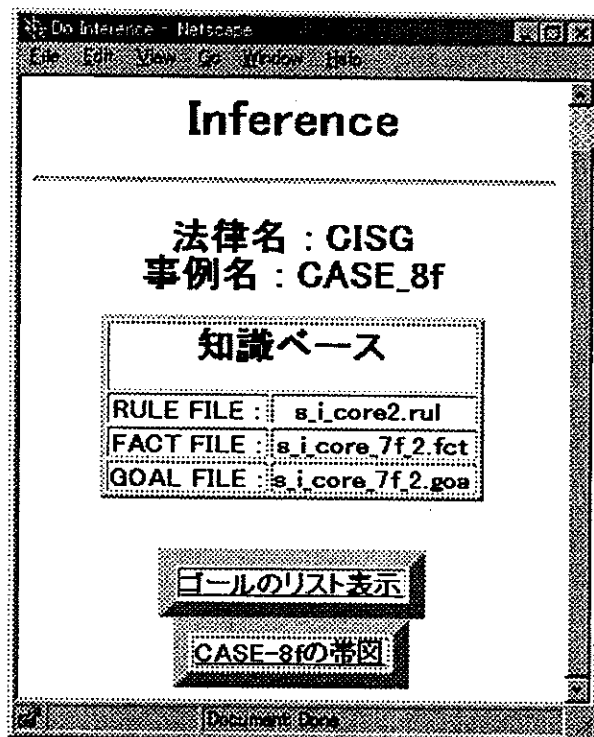


図 5: ゴール入力方式の選択

2.4.3 ゴール入力方式の選択

次にゴール入力方式の選択画面 (図 5) が表示

される。ここには確認のために、ルール、ファクト、ゴールのファイル名が表示されている。「ゴールリスト表示」をクリックすれば、ゴールファイルの中の実行できるゴールの一覧が表示される(図6)。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、その結果と根拠が説明画面として表示される(図9)。

2.4.4 帯図表示

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、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った。これを、HTMLの表形式記述を用いて実現したのがゴールの帯図表示である(図7)。帯図は各時点でのアンカーを選択することで説明機能呼び出すことが可能となっている。帯図の中の法的義務や権利を表す文をクリックすると、その法的効力が発生した時点(始点)と終了した時点(終点)を選択する画面が表示される(図8)。これらの2つのボタンのうち一つをクリックすることにより、それぞれを証明するために用いられたゴールの説明推論が起動され、推論結果の説明が表示される。

2.4.5 推論結果表示

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。この画面では、適用に成功したルール、適用して証明されたゴール、その適用において成功したサブゴールを表示する(図9)。適用されたルールは下線のついたルールIDとしてアンカー表示され、これをクリックするとそのルールの内容が表示される(図11)。その下に、適用して証明されたゴールが自然言語変換されて表示される。さらにその下の「BE-



図 6: ゴール一覧

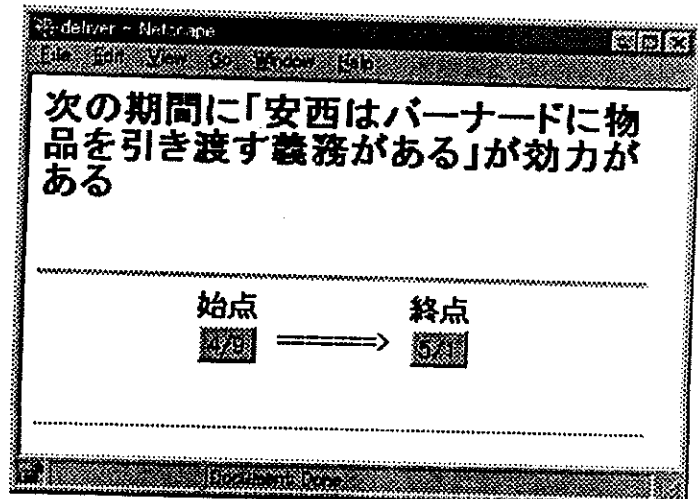


図 8: 始点と終点の選択画面

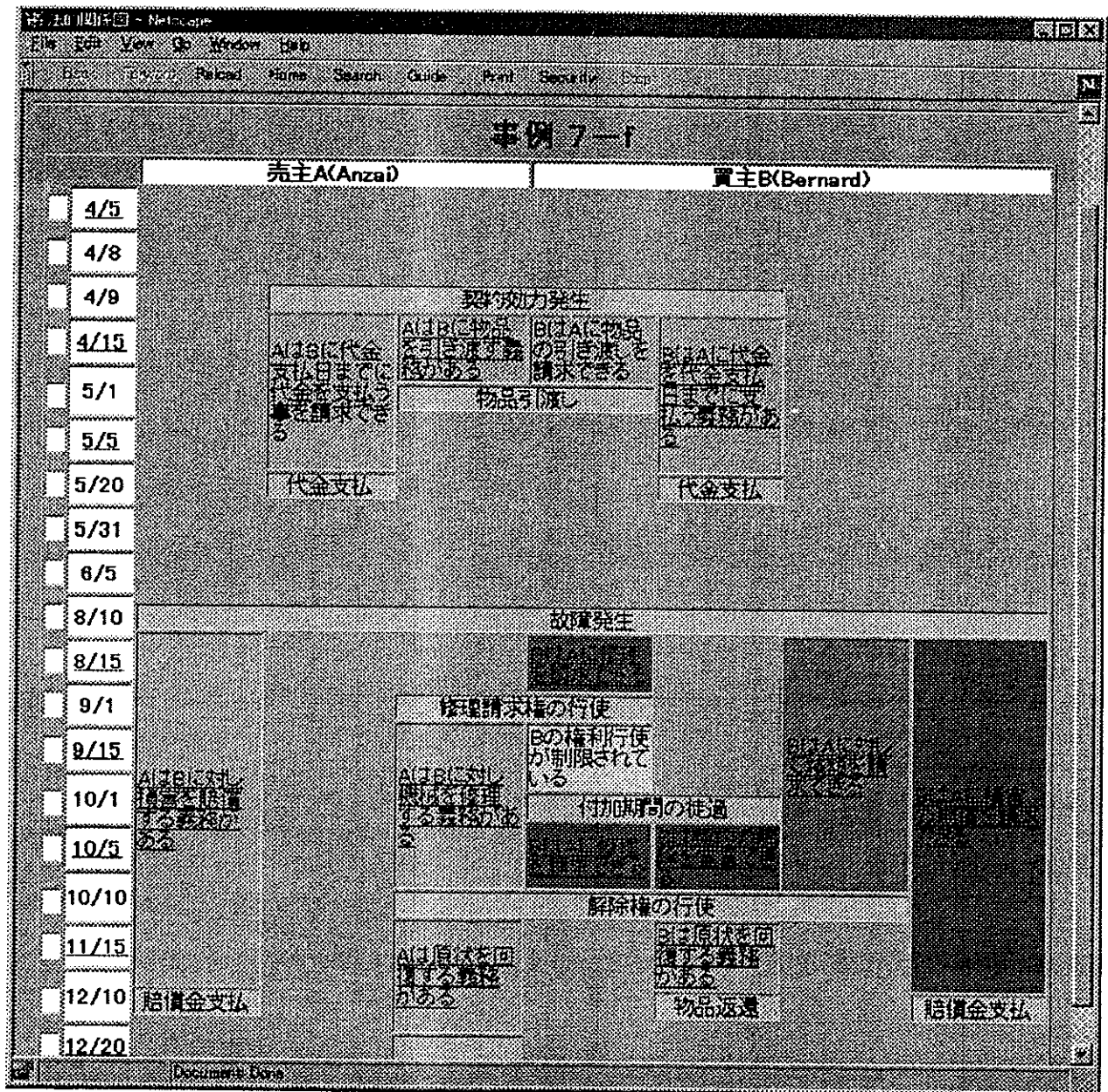


図 7: 帯図表示画面

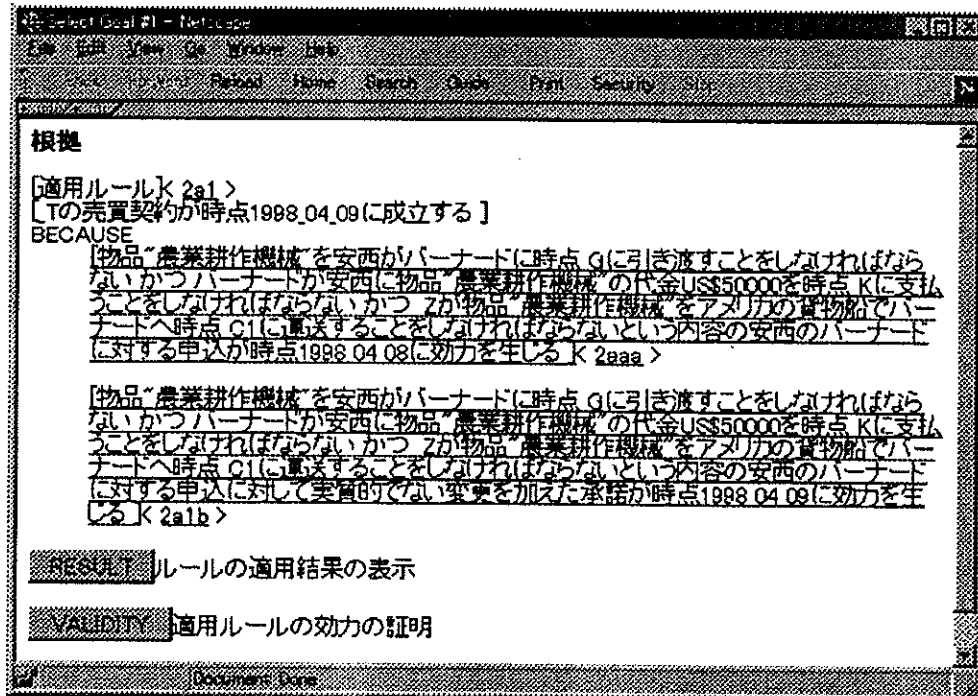


図 9: 推論結果表示説明画面

CAUSE」以下に表示されているのが、成功したサブゴールである。ここで表示されるのは、ゴールを証明するのに使用された証明木のうち成功したサブゴールだけである。このサブゴールはアンカーになっており、これ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。

また、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得られるようになっている。

画面の一番下には 2 つのボタンが表示されている。「RESULT」ボタンは証明されたゴールにおける変数の束縛状況をルールにあてはめた状態を表示する。これを押すと、「ルールの適用結果表示画面 (図 10)」を表示する。「VALIDITY」ボタンは、メタ推論により、このルールの効力の証明を行なうボタンである。

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 13 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。

2.4.6 ルール表示

ルールの表示画面では、図 11 に示されるように、そのルールの説明文章および、関連情報を表示するためのアンカーが上部に表示される。このアンカーをクリックすれば、HTML で記述された関連情報の表示に飛ぶ。この例では、[CISG46] と示された部分をクリックすると、国際売買法第 46 条の条文が図 12 のように表示される。この条文は通常の HTML ファイルで記述されているので、そこからさらにインターネット上の情報ソースへのリンクを貼って置くことも容易である。知識ベース内のルールとの関連付けは、ルール毎にファイル名がファクトとして記述しておくだけでよい。

また、表示画面の下部では、CPF で書かれたルールの構造を、自然言語変換システムおよびプリティプリント機能により入れ子形式で表示する。これにより CPF の論理構造を一見で把握することが可能になっている。

2.4.7 ファクト表示

ファクトの表示画面では、図 13 に示されるように、適用されたファクトを自然言語変換さ

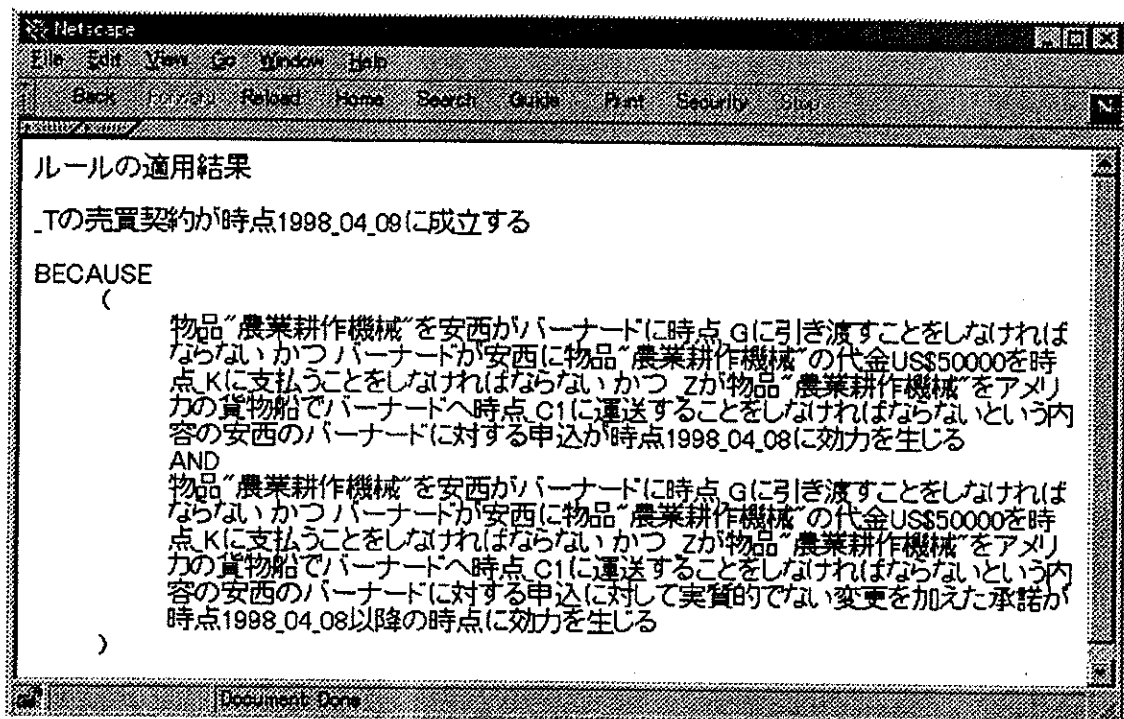


図 10: ルール適用結果表示

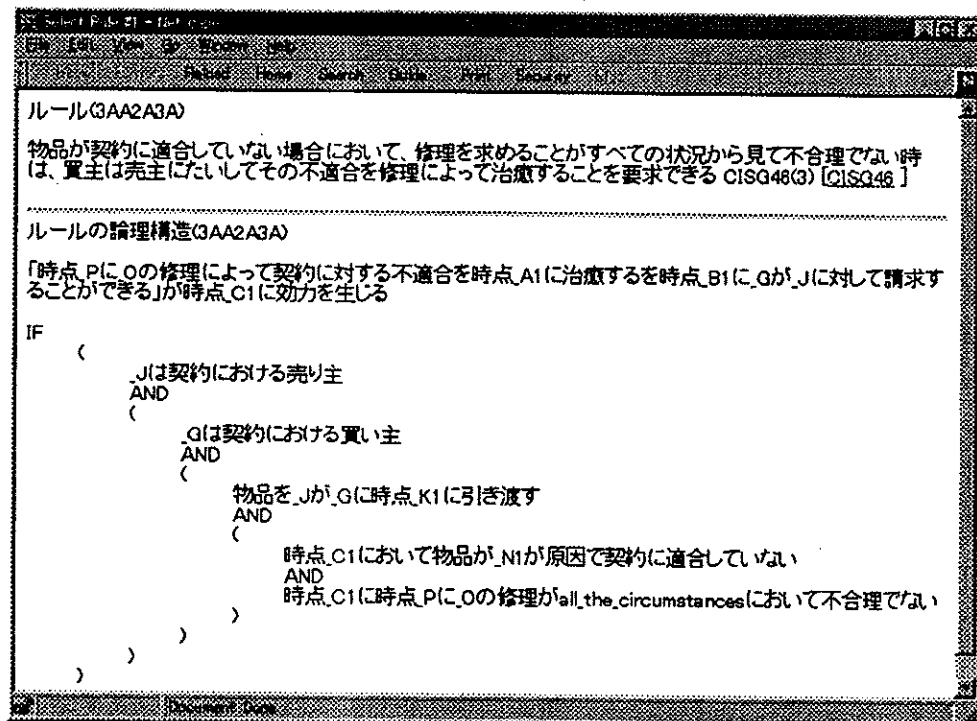


図 11: ルール表示

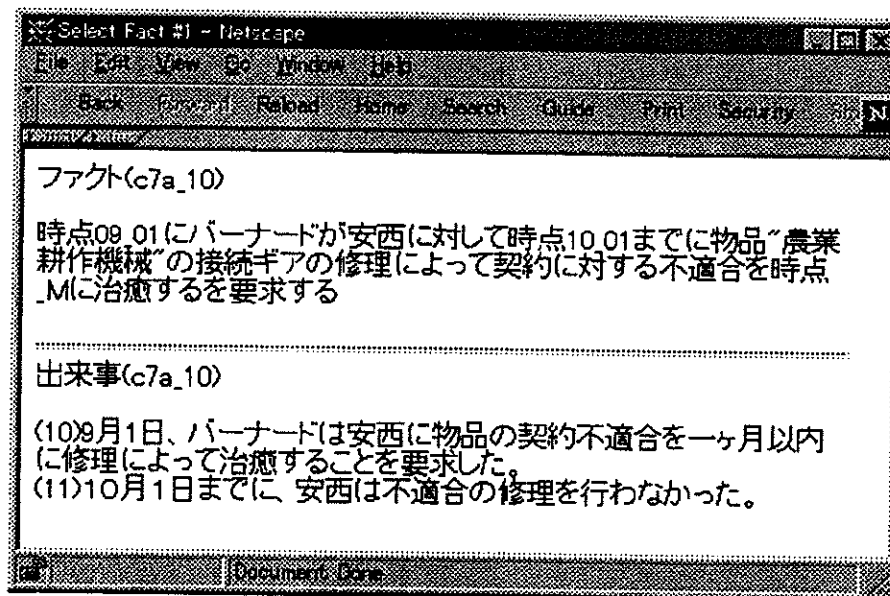


図 13: ファクト表示

れたものが表示される。また、このファクトを含む出来事についての説明文が表示される。この情報は知識ベースにテキストとしてあらかじめ書き込まれている。

2.5 機能制約と拡張性

2.5.1 CPF デバッガとしての機能

本推論説明実行システムは、第3節で述べる知識ベース作成支援システムで作成しデバッグした知識ベースを実行するためのシステムとして設計されており、知識ベースのデバッグのための機能はもたない。例えば、今回実装した成功ゴールの説明機能は一種のステップアとして働くが、説明機能として実現したために、成功したサブゴールの表示に表示を限定している。これにより、推論実行内容の説明に対しては有用である反面、実行時情報を割愛しているために、知識ベースのデバッグには不向きである。将来の拡張可能性として、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考えられるが、説明機能との両立のためのインタフェースの設計などが課題である。

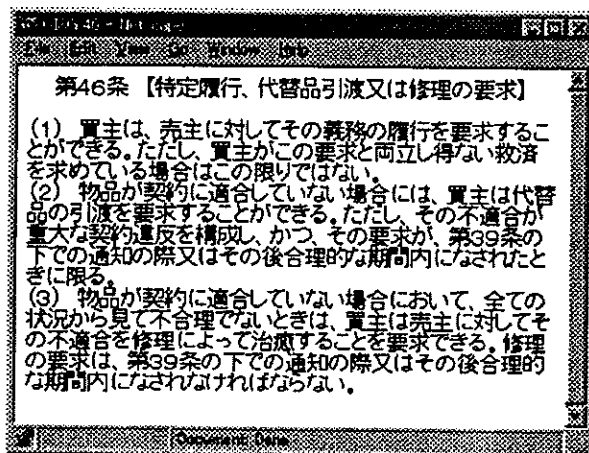


図 12: 条文の表示

2.5.2 複数のサーバの同時実行

開発した推論説明実行システムは、複数の異なる推論プロセスを並行して実行するには設計されていない。これを実現するためには、複数サーバプロセスの管理をユーザ毎におこなうメカニズムを開発するなどの必要があり、本研究の期間では実現できなかった。ただし、本システムは WWW サーバを用いて実現されており、かつゴールの実行を独立の CGI 呼び出しとして実現しているので、同じ知識ベースを実行している限りは、複数の異なるゴールの実行を同一のサーバに対して行なうことができる。

2.5.3 失敗の説明

ゴールの実行が失敗した時の説明機能として、本システムは、失敗の説明機能を一部実現している。しかし、成功の説明と異なり、失敗の説明は、推論過程で試したすべてのサブゴールの組み合わせをうまく表示させなければならないため、どのようなタイミングでどのような形式で表示すべきかの設計が大変難しい。使いやすいものにするためには、まだ、一層の努力が必要である。

2.5.4 メタ推論

本システムは、推論に使用したルールの効力を証明する機能としてメタ推論を一部実現している。しかし、推論ルールとメタルールのルール記述方式や、説明機能の表示のタイミングの設計など難しい問題点が多くあり、実用インタフェースとしてはまだまだ問題が多い。今後の課題である。

2.5.5 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ただけであ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。

3 知識ベース作成支援・推論実行システム LES-5

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 14～図 16) 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 14 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、図 15 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 16 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

Knowledge Editor Version 1.0			
Change Text	Dictionary	Fact Editor	Language
FILE	EDIT	SEARCH	HELP
Terminal Representation 1 < q5grsy auc > (1:1)			
goal('is concluded',[act:'AGT_CONTRACT, obj:'contract',[agt:'AGT_CONTRACT, cnt:'CNT_CONTRACT, inp:'DPF_CONTRACT, obj:'OBJ_CONTRACT]), tim:T1))			
Textual Representation 1 < q5grsy auc > (12:1)			
participle_place time_after time_before time_of_concomitance transmission また、sav は2つあるが、抽の数が正しいのか？ in_table('accept',[agt:'AGT_ACCEPT,obj:'OBJ_ACCEPT,tim:T1M_ACCEPT]), [[[時点,T1M_ACCEPT,'に'),(AGT_ACCEPT,'が'),(OBJ_ACCEPT,'を')], 承諾する]]) in_table('acceptance',[agt:'AGT_ACCEPTANCE,cnt:'CNT_ACCEPTANC]) dicvui 監修書 1995/11/19(Fri) Header /home/swede/mec/works/ad.dtc.v 1.3 1996/01/12 15:38:11 swa de Exp swede dic(dlc, (e,[is_obligatory],[[concept{[goe,obj,tim]:[]}], super_sub_concept([sup:[],sub:[],syn:[],ant:[]]))).			
Dictionary Editor < q5dic auc > (1:1)			
goal('is concluded',[act:'AGT_CONTRACT, obj:'contract',[agt:'AGT_CONTRACT, cnt:'CNT_CONTRACT, inp:'DPF_CONTRACT, obj:'OBJ_CONTRACT]), tim:T1))			
Rule Editor < q5ruly auc > (13:8)			
3 義務の存在 revised at 96.08.01, 10.05 san('g',[(3)時点TにX義務がある X is obligatory for A at time T(in legal world) is_obligatory (IS_OBLIGATORY,[goe:A, obj:X, tim:T]) < 31 時点Tに「X義務がある」が効力がある X is obligatory for A is valid at time T. is_valid (IS_VALID,[obj:san(SENG),['is obligatory' (IS_OBLIGATORY,[goe:A, obj:X, tim:T])), goe:A, tim:T]))), goe:A, tim:T]))< XXXXXXXXXX is obligatory と must を結び付けるルールを作った ルの義務がある」と「ねばならない」を結び付けるルールAdded by HY on 1995/12/ 4(Mon) 16:12 以下のルール(31)は is obligatory < must という階層辞書にしたがってル- ルの具体が行われるといふことになる。 31 'is_valid (IS_VALID,[obj:san(SENG_IS_OBLIGATORY,['is obligatory (HAS_DUTY,[goe:A, obj:X, tim:T])), goe:A, tim:T]))<	3 義務の存在 revised at 96.08.01, 10.05 san('g',[(3)時点TにX義務がある X is obligatory for A at time T(in legal world) is_obligatory (IS_OBLIGATORY,[goe:A, obj:X, tim:T]) < 31 時点Tに「X義務がある」が効力がある X is obligatory for A is valid at time T. is_valid (IS_VALID,[obj:san(SENG),['is obligatory' (IS_OBLIGATORY,[goe:A, obj:X, tim:T])), goe:A, tim:T]))), goe:A, tim:T]))< XXXXXXXXXX is obligatory と must を結び付けるルールを作った ルの義務がある」と「ねばならない」を結び付けるルールAdded by HY on 1995/12/ 4(Mon) 16:12 以下のルール(31)は is obligatory < must という階層辞書にしたがってル- ルの具体が行われるといふことになる。 31 'is_valid (IS_VALID,[obj:san(SENG_IS_OBLIGATORY,['is obligatory (HAS_DUTY,[goe:A, obj:X, tim:T])), goe:A, tim:T]))<		

図 16: 知識エディタ画面

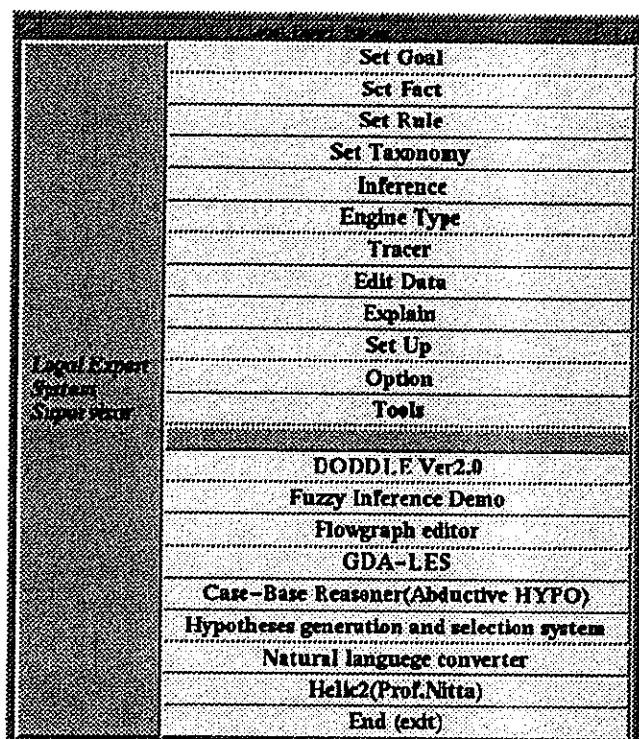


図 14: システムメインメニュー画面

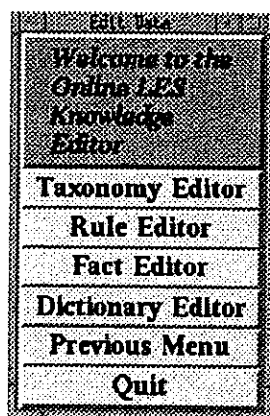


図 15: エディタメニュー画面

して、トレース機能、スパイ機能、ビジュアルトレーサ機能を持つ。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 17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。

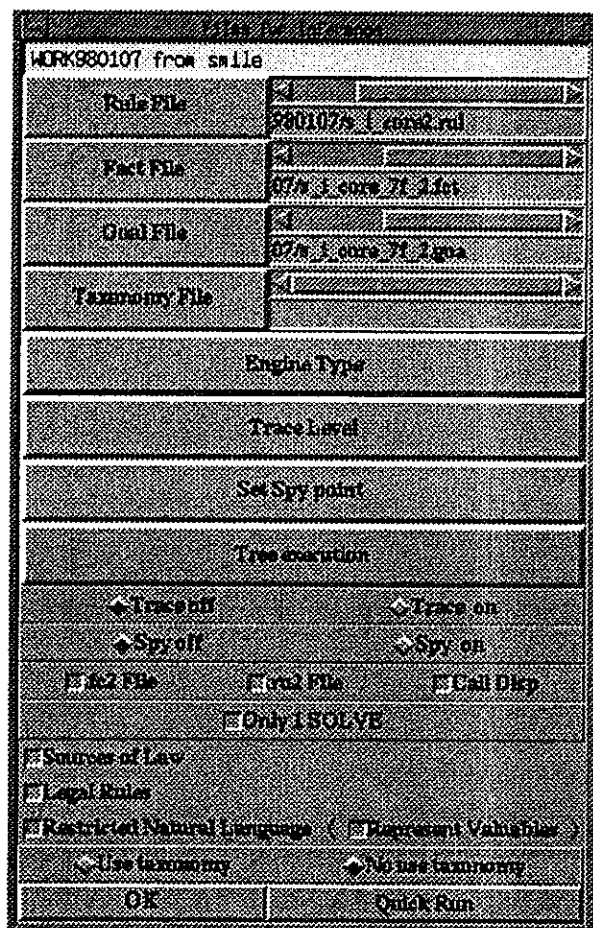


図 17: 推論実行ダイアログ

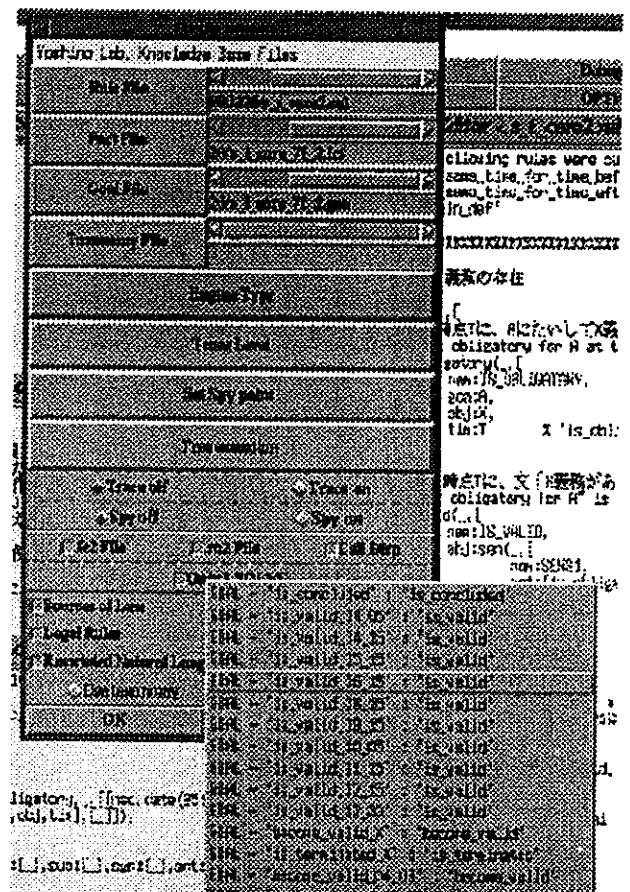


図 18: ゴール実行

- 「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 18）。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 19）
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

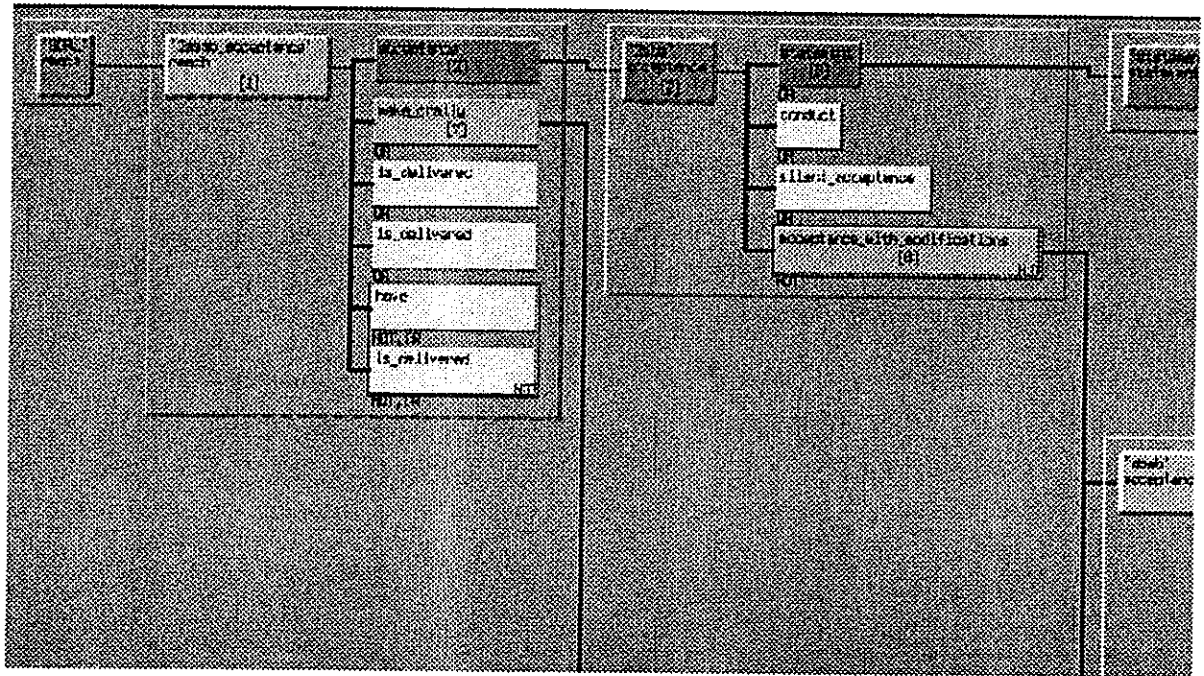


図 19: ビジュアルトレース画面

- 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

```
'is_obligatory'(IS_OBLIGATORY,[
    goa:A,
    obj:X,
    tim:T])
]),
goa:A,
tim:T
]).
```

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。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,[
```

<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 20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、該当部分が登録されている変換テーブルを通して自然言語に変換され表示される。

図 21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 22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。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。

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、以下のような知識ベース作成のための機能も備えている。

3.3.1 辞書作成機能

辞書を作成するための支援機能として、辞書作成機能が用意されている。辞書作成機能を起動すると、図 23のような辞書作成メインダイアログが表示される。辞書 ID は全辞書中でユニークな任意の文字列を与える。

1. 「STRUC」ボタンをクリックすると下記のような内容を持った、Structure ダイアログが表示されるので、各項目を入力して「OK」ボタンをクリックする。

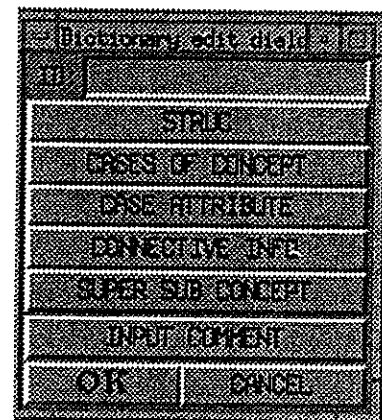


図 23: 辞書作成メインダイアログ画面
LANG 現在未使用

WORD 述語名 (必須)

CONCEPT コンセプト名 (必須)⁴

GRAMMER 文献 (任意)

WRITER 入力者 (自動記述)

DATE 入力年月日時刻 (自動記述)

INPUT COMMENT コメント (任意)

2. 「CASES OF CONCEPT」ボタンをクリックすると格指定ダイアログが表示される。「Cases list」ボタンで格リストが表示される。格リストの格をダブルクリックすると格が選択される。⁵
3. 「CASE ATTRIBUTE」ボタンをクリックして格の変数名を入力する。デフォルト値が用意されている。
4. 「CONNECTIVE INFO」ボタンをクリックして、自然言語変換時の格の接続詞情報を入力する。
5. 「SUPER SUB CONCEPT」ボタンをクリックして、述語の上位概念、下位概念、同義語などの情報を入力する。

⁴ほとんどの場合は WORD と CONCEPT には、同じ記述が入る。

⁵補助格は現在未使用

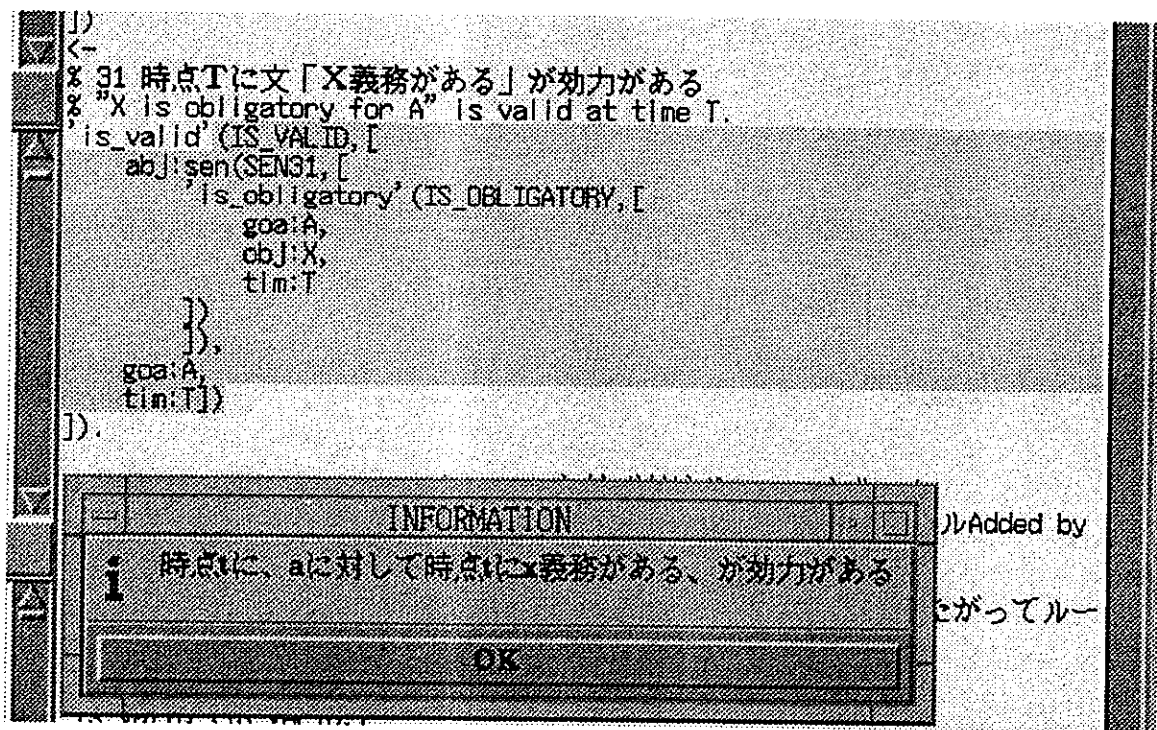


図 20: ルールの自然言語表示

6. 全ての情報を入力したら、メインダイアログの「OK」ボタンをクリックすると、辞書ウィンドウ画面に新しい辞書が挿入される。

「CASE ATTRIBUTE」、「CONNECTIVE INFO」、「SUPER SUB CONCEPT」は現在のところは任意の入力項目である。

3.3.2 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],[_]]),
-,
-,
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
])
```

次のような手順でこのルールタームの生成機能が利用できるようになっている。

1. 必要なら「Dictionary search」で辞書の検索を行う。
2. 辞書エレメントの「dic(」という文字をダブルクリックして選択する (図 24)。

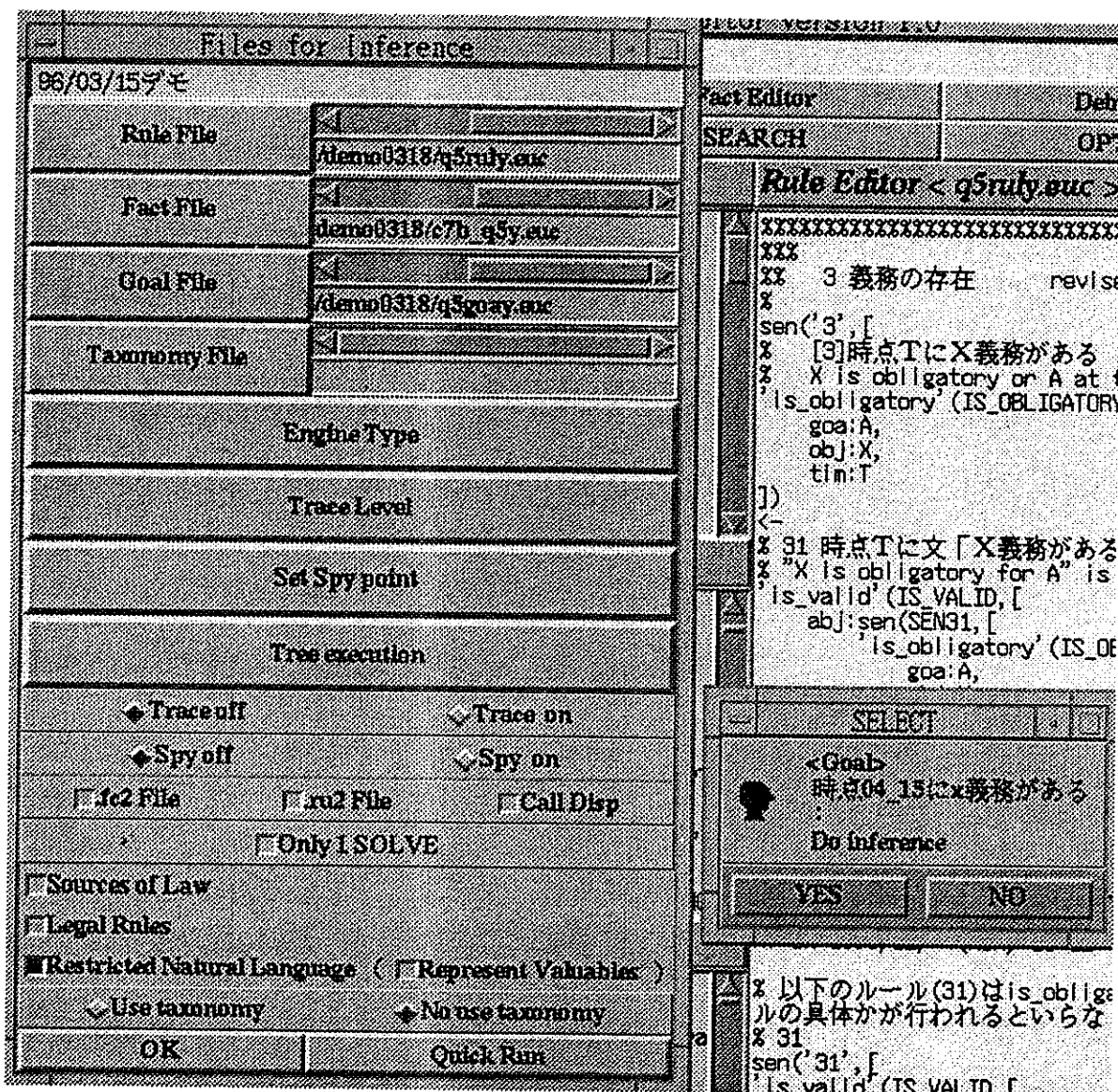


図 21: ゴールの自然言語表示

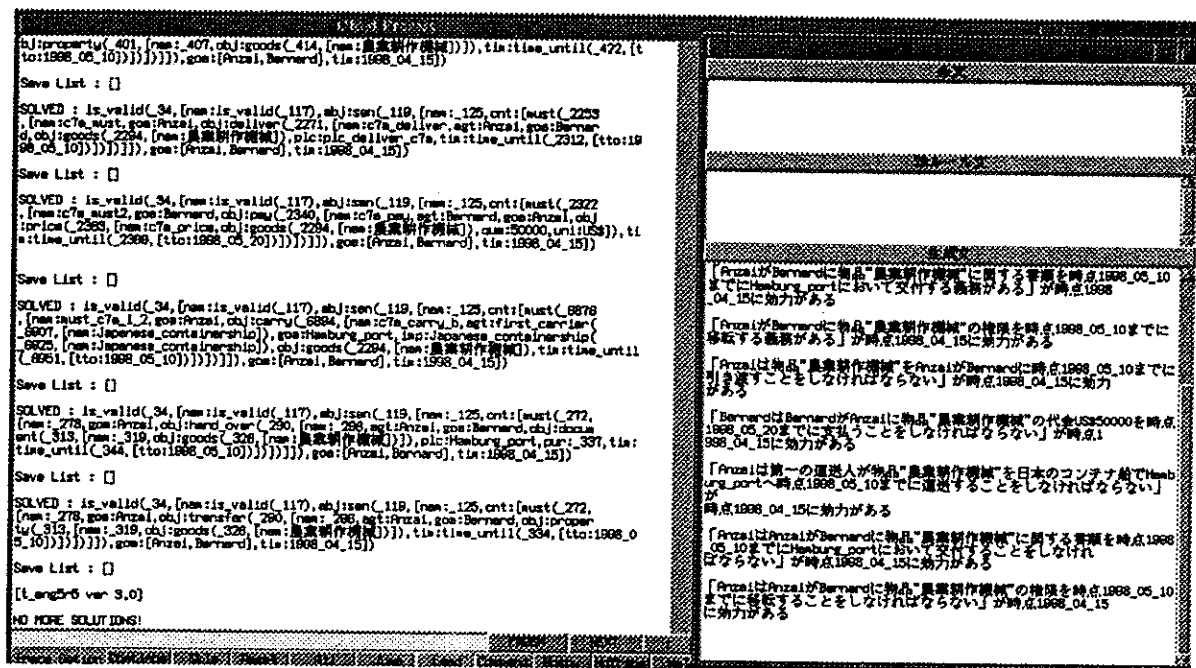


図 22: 推論結果の自然言語表示

3. CPF を挿入するウィンドウにフォーカスを移して、挿入場所にカーソルを移動する。
4. 「Apply dictionary to rule」をクリックすると、カーソル位置に辞書から生成された CPF が挿入される。

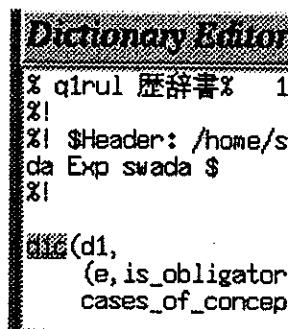


図 24: 辞書の選択

3.3.3 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 25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 25のようなチェッカーウィンドウが現れて、シンタックスチェックを行う。

3.3.4 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。

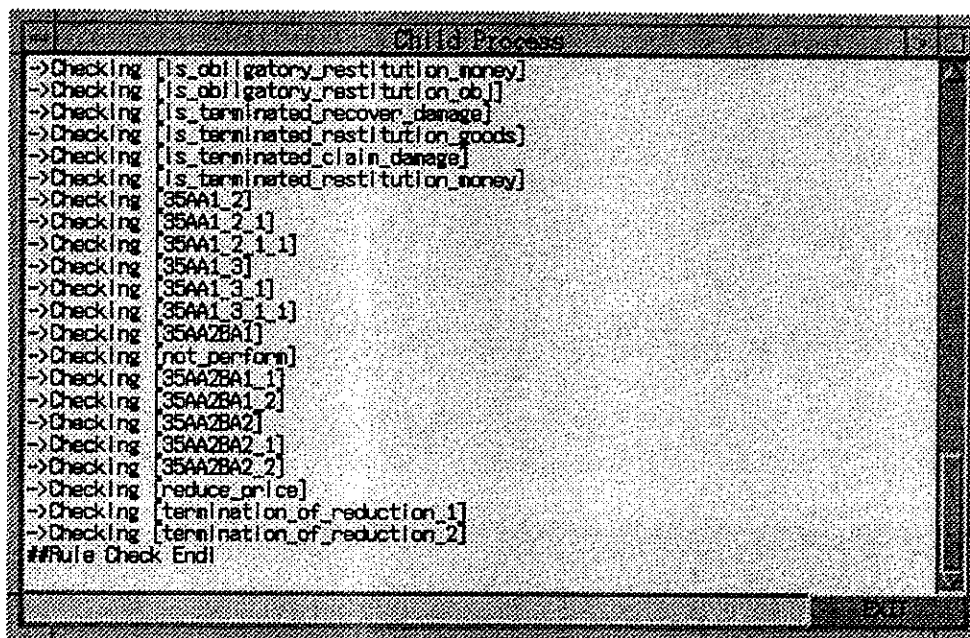


図 25: 文法チェック機能

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。