

Prologを用いた秘書エキスパートシステム

東田 正信 和佐野 哲男 森原 一郎
(日本電信電話公社 横須賀電気通信研究所)

1. はじめに

知識ベース技術を適用した代表的な応用システムには

①自然言語理解システム

②エキスパートシステム

がある。

①は高度の技術を必要とするものであり、それ自体が大きな研究項目になっている。そのため、現在までに試作・開発されたほとんどのシステムはデータベース検索を行うものに限られている。⁽¹⁾など)

②は専門家の知識を利用した問題解決のためのシステムであり、ほとんどのシステムが自然言語インターフェースを持たず、メニュー選択またはシステムからの質問に答えるというシステム主導型になっている。

しかしながら一般のユーザーにも使いやすい実用的なエキスパートシステムとするためにはシステムとの対応は自然言語で行えることが望ましい。また自然な感じでシステムを使えるようにするために、必要情報を全て含む入力文を作成する煩わしさからユーザーを解放する必要がある。

従って今後は①と②が融合されたユーザー主導型のエキスパートシステムの開発の必要性が増してくるものと考えられる。そのためには不十分な情報しか持たない入力を許容し、自然な対話を通じた適切な問題解決機能を実現することが必要である。

このような観点から、自然言語のインターフェースを必要とし、知識ベース技術が最も早期に導入される可能性の高い〇

Aの分野の中から、多くのノウハウを蓄積する必要のある秘書業務を選択し、各種日程表の作成・調整に関する知識を知識ベース化した知的秘書システムシステムを試作した。⁽²⁾

従来、この分野での自動化はOdyssey³⁾やNUDGE⁴⁾システムなどの試みがあるが、それぞれ自然言語インターフェースを持たない、レスポンスタイムが遅いなどの問題があった。

本システムは上記のような問題を解決することを目的に試作を行った。本システムは簡易な日本語のインターフェースを持っており自然な会話を通じて日程等の検索の他、作成・変更などの一連のデータベース操作の実行が可能で、秘書業務の自動化を図ることができる。

2. システムの概要と構成

本システムでは以下の機能を実現できることを目標とした。

- (1) 自然語入力が可能で、曖昧入力・省略語入力を許容し、自然な会話を実現する機能
- (2) 知識ベース化された秘書のノウハウを用いた推論により各種日程の自動作成・自動調整を行う機能

これらの機能を実現するためにシステム構成は図2-1のように①日本語処理サブシステムと②問題解決サブシステムの2つのサブシステムから成っている。

本システム全体のモジュール構成を図2-2に示す。

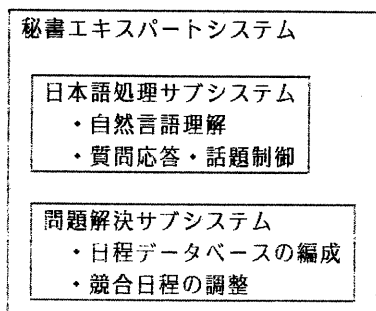


図2-1 秘書エキスパートシステムの構成

このシステムは知能処理言語DEC-10Pr
olog⁵⁾を用い、DEC システム2060上に構
築した。尚入力将来音声入出力装置
と結合されることを考慮してローマ字入
力とし、出力および入力のエコーバック
を漢字かな交りの日本語文で表示するこ
ととした。

本システムを使用した時の会話例を図
2-3に示す。

以下に知識ベースの構成・知識表現法
・推論方法などについて述べる。

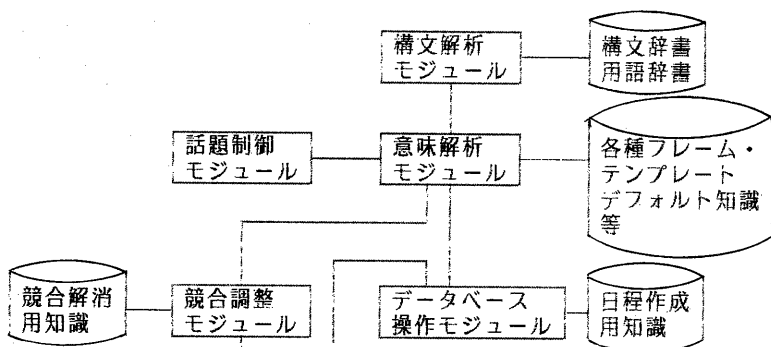


図2-2 秘書エキスパートシステムのモジュール構成

U : 来週の金曜日の午後1時から幹部会議を開きたい。 S : 何時までですか? U : 5時までです。 S : 4月6日(金曜日)の13時から17時までですね。 社長の予定は空いていますので出席はOKです。 A専務は2時から3時まで面会ですが変更してもいいですか? U : はい。 S : D専務は4時から6時まで外出ですが変更してもいいですか? U : はい。 S : E専務は1時から4時まで部長会議ですが変更してもいいですか? U : いいえ。 S : 規定された定足数に足りましたので幹部会議を登録します。 出席者は社長、副社長、A専務、B専務、C専務、D専務、 F専務です。欠席者はE専務です。	(システム内動作処理) ・不足情報の問合せ ・曖昧情報の除去 ・ルール”社長の出席は必須”を満足 ・ルール”定足数は6人”が満足されないため調整を実行 ・調整後定足数に足りたので会議を登録
---	---

U : ユーザー、S : システム

図2-3. 秘書エキスパートシステムとの会話例

3. 知識の種類と表現方法

3.1 知識の種類

本システムで知識ベースとして扱う知識を以下に示す。個人日程・行事日程はデータベースとして考えることとした。

①構文解析知識

- ・構文則
- ・辞書

②意味解析知識

- ・キーワード抽出知識
- ・デフォルト知識
- ・派生データ生成知識

③事務・人事知識

- ・カレンダー知識
- ・人物・役職知識
- ・行事知識
- ・地名知識

④業務処理知識

- ・業務処理手順知識
- ・スケジュール調整知識

これらの知識はPrologのプログラムとして記述されるが知識の編集の可能性・変更等の容易性などを考慮して以下の三種類に分類して実現した。

3.2 知識の分類

(1) システム使用環境下で変化することのない知識（アルゴリズム的な知識）

恒等的に真であると考えられる知識、変更されることの無い知識、固定的な処理の流れを表わす知識などでPrologのnon-unit clause、すなわちプログラムの処理の中に組み込まれた知識として表現する。（キーワード抽出・カレンダー・派生データ生成知識等）

（例）明日を定義する知識
asu([Y, M, D], [NY, NM, ND]) :-

```
( ND is D+1, NM=M, NY=Y;  
  NM is M+1, ND=1, NY=Y;  
  NY is Y+1, ND=1, NM=1).
```

(2) システムの使用状況に応じて内容が変化する知識

システムの使用過程で作成、登録、修正、追加、変更等の作用を受ける知識でPrologのunit clause、即ちassertionとして表現する。

ここでは知識の内容に応じて以下の二種類の知識として分類する。

①フレーム型知識

事実に関する知識、データの知識であり、この知識はそのままPrologのプログラムとして組み込まれて実行される。（人事・行事・地名知識等）

（例）

```
buchou(eigyuu, satou).  
kyou([1984, 7, 12]).
```

②ルール型知識

手順、方法等を表現する知識であり、プロダクションルールの形式で表現することとし、条件部、実行部に記述された内容はPrologで書かれたルールインタプリタによって解釈実行する。ルールは削除、追加や適用順序の指定などが可能な様に番号を付与する。（スケジュール調整知識等）

（例）

```
(rule16, if, (c1, c2, ..., cn),  
  then, ( a1, a2, ..., am)).
```

3.3 データベースの形式

個人日程のデータおよび行事日程のデータは以下に示す形式で格納し、個人単位・行事単位でファイル化する。

(例)

- ・個人データ
day-sch(氏名, 年月日, 30分単位
の予定).
- ・行事データ
kaigi-sch(会議の種類, 開催月日,
時間, 場所, 出席者, 議題)

4. 日本語処理システム

本システムの日本語処理に対する要求
条件は実用性の面から

- ① 処理速度が速いこと(レスポンス
タイムが許容限度内であること)
- ② ユーザーの要求を実行する処理の
ために必要な情報を提供できること
が必要である。

①に対してはユーザーが使用する表現
を調査し、実際にシステムを使用する上
で不便を感じない程度の制約を設けて高
速化・効率化を図ることとした。

- ・入力文は単文とする
- ・否定文は扱わない
- ・入力文にはユーザーの要求(データ
ベース操作内容)が連想できる語句
を含んでいること。

適用分野を秘書業務に限定したことで
意味上の曖昧さ(同じ単語が複数の意味
を持つ)を含む文はほとんど無いと考え
られる。また構文上の曖昧さを含む文
([例] AさんはBさんとCさんに会う)
はどちらかの意味に限定した解釈を行い、
結果がユーザーの意図と異なる場合は改
めて対話的に修正することとした。

以上の環境を設定することによって日
本語処理の処理効率を大幅に改善するこ
とができた。

日本語処理の手順としては従来から各
種システムで採用されている構文解析と
意味解析の二段構えの構成としたが、意
味解析では必要となる知識を知識ベース
化し、これを利用する方法を導入した。

4. 1 構文解析

構文解析に用いた手法はDEC-10Prolog
の中で提供されている文脈自由文法(Con
text Free Grammar)の発展形であるDC
G(Definite Clause Grammar)を使用した。
したがって構文解析により作成された構
文木から分解された各単語の品詞と文法
的な役割(主語・目的語など)が明らか
になる他、辞書の構成を工夫することに
より後の意味解析に必要な情報を構文解
析結果の付加情報として付与できる。

特に同じ助詞でも場所を表わす場合と
時間を表わす場合があるため、これらを
区別する情報を付加し、次の意味解析で
利用できるようにした。また動詞・助動
詞の活用形については意味情報を余り持
たないと考え、構文木を深くすることを
避けるためにこれらの活用形はすべて辞
書として保持することとした。

構文解析の結果例を以下に示す。

[例文1]

『部長 は 明日 の 午後
主語名詞 助詞 時間語 助詞 時間語

1 時 から 8
数詞 時間語 時間助詞 数詞

時 まで 東京大学
時間語 時間助詞 場所名詞

へ 行く。』
場所助詞 動詞

このように構文則は時間や場所を表現
する語句およびそれらの役割を決める助
詞に関する情報がきめ細かく解析できる
ものとしており、その数は約100であ
る。

4. 2 意味解析

文の構造を扱う構文解析については広く研究されていて、手法も確立されてきているのに対し、意味解析は、論理式に基づくモンタギュー文法からFilmore の格文法、Schankの概念依存文法などの一種のフレームに基づく文法などが考えられているものの決定的な手法がないという状態である。⁶⁾

一般に意味解析が構文解析に比べて格段に困難とされる理由は以下のものがある。

- ①意味解析に必要・十分な知識が不明である。
- ②同一の意味を表現する多くの表現方法（言回し、言替え）が存在する。

①に関しては対話の中で我々がごく当り前と考えているような事（常識）に対する応答の可否によってシステムのきめ細かさが大きく左右されるため、いわゆる常識（カレンダー、祝日、地名、事務などに関する知識）とされている知識を知識ベース化してこれに対処することとした。

②に関しては単語の場合は異なる表現であっても同義であることを構文辞書の中で持ち、言回しの場合は規格化した表現に変換することで対処した。

また本システムのように適用分野を限定することで構文上・意味上の曖昧さをかなり除去できることを前述したが、この制約はこの他にユーザーの意図をシステム側で予測し、その予測に沿って必要な情報を収拾していくことができるという利点を持っている。

本システムの意味解析は上記の様な背景から格文法を採用し、ユーザーの要求をフレームの形式で表現し、知識ベースを利用した意味解析によってフレーム内の各スロットに必要な情報を取込んでいく手法を採った。この処理を以下の①から

④の手順で行う。

①要求キーワードの抽出

ユーザーからの要求は個人または行事日程に関する照会・登録・変更・消去のうちのいずれかである。従って入力文の中からこれらの内のどれであるかが判断できる用語を抽出する。

（例）

～は……検索
～へ行く……登録
～は～に変える……変更
～は中止する……削除

②入力フレームの作成

入力文からそれがどんな行事・日程に関する要求であるかを判断してそれに対応するフレームを作成する。これを入力フレームと呼ぶ。必要情報は構文解析の結果得られる句構造の情報と各単語に付与された情報をもとにして抽出する。例文1に対する入力フレームの構造を図4-1に示す

入力フレーム	
作業種別	登録（行事）
動作	行く
誰が	部長
誰に（を）	NIL
何を	NIL
何に	NIL
何時から	59.7.13,13:00*
何時に	59.7.13,20:00*
どこから	NIL
どこへ	東京大学
何故	NIL

*今日が59.7.12 の場合

図4-1 入力フレームの例

この段階で”明日””午後1時””8時”などは知識ベースを用いて省略を補完し、正規化された表現に変換すること

で曖昧さを除去する。

ここでNIL は入力からは得られなかった情報であることを示す。

③行事フレームの選択と補完

次に入力フレームから対象となる行事のフレームを選択し、入力フレームの内容と比較して必要な情報となくともよい情報を区別し、後者には入力フレームにNOを書き込む。

行事フレームにデフォルト値として登録されているものがあればそれを使用する。

ここでは動作が“行く”でであったために入力フレームの“何を”の項がNILになっていて対象行事が分らないが、このような場合には以下の様な場合分けを区別する係受けの知識を用いて補完する。(例文1で“東京大学へ出張する”となっていればこの必要はない。)

東京に行く………出張(地名、場所)
ゴルフに行く……レジャー
委員会に行く……会議(会議名)

この結果入力フレームは図4-2のようになる。

入力フレーム	
作業種別	登録(行事)
動作	行く
誰が	部長
誰に(を)	NO
何を	出張
何に	NO
何時から	59.7.13,13:00*
何時に	59.7.13,20:00*
どこから	NO
どこへ	東京大学
何故	NIL

図4-2 補足された入力フレーム

ここでまだNIL である項目はQA等による情報收拾が必要な項目となる。

④話題ベースによる補完

最後に話題ベースと呼ばれる会話の流れを制御している知識ベース(後述)を利用して入力フレームの補完を行う。この操作を終えた後も残るNIL についてはデータベース操作の段階で必要であればユーザーにQAで問合せて情報を收拾する。

この様にして得られた入力フレームが入力文の意味解析の結果としてデータベース操作部に送られる。

5. 問題解決処理システム

従来の日本語質問応答システムではユーザーの入力文の解析=データベース検索コマンドの生成であることが多いため、入力文には検索に必要な情報がすべて含まれていなければならなかったが、本システムのように曖昧入力・省略語入力を許容し、データベースの検索以外にも登録・変更・消去などの操作をする場合には以下のような問題が新たに生ずる。

- (1) 入力フレームに操作に必要な情報がすべて含まれていない時の処理
- (2) 入力フレームの指示を実行する時に競合が発生する場合

(1) に関しては必要情報をユーザーにQAを通して聞くことで基本的には対処可能だが、以下の様な不必要な部分を含む会話を避けるためには工夫を要する。

u. 明日の会議はあさっての午後3時からにする。

s. 何時までですか? 不必要

u. 5時まで。

s. ok.

明日に指定された会議はありません。

この問題に対処するために以下の手順で処理を進める。

- ①入力フレームに従ってまず該当するデータフレームの有無を検索する。
- ②一意に同定できない時はユーザーに聞く。一意であって処理が続行できるときは続けて実行する。

(2) に関しては競合状態を解決するノウハウをシステムが知識ベースとして持っていて、それに基く推論によって処理を進めていく必要がある。

この知識はこのシステムが使用される環境に依存することから知識の分類のうちで(2)の②のルール型の知識と考え、プロダクションルールとして表現し、ルールの追加や削除が可能にようにした。したがってこれらのルールはルールインタプリタ(推論機構)で解釈実行させる。ルールインタプリタはProlog in Prologの考え方に従って作成した。

本システムで複数の参加者がいる会議の登録のように出席者のリストを調べ、その人達の時間の調整をする必要のある場合のルールを例として示す(理解が容易な様に条件・実行部を日本語で書く)。

[ルールの例]

- (rule1, if, (出席予定者の時間帯に競合がない), then, (会議を登録), (出席予定者の日程を更新)).
- (rule2, if, (議長の時間帯に競合がある), (議長の出席不可), then, (会議の登録中止)).
- (rule3, if, (議長の時間帯に競合がある), (議長の出席可), (出席可能者の数が定足数に足りない), then, (会議を登録中止)).
- (rule4, if, (議長の時間帯に競合がある), (議長の出席可), (出席可能者の数が定足数に足りる),

then, (会議を登録), (出席可能者の日程を更新)).

ここで条件節・実行節の中にある項の具体的な表現は以下の用に定義され、Prologの導出原理に従って実行される。

(議長の出席可): -

(議長の名前を調査),

(議長に競合予定の変更の可否を問合せ), (返答がYES).

(議長の出席不可): -

(議長の名前を調査),

(議長に競合予定の変更の可否を問合せ), (返答がNO).

現在この他に時間のシフトや予定日の移動などに関するルールがある。これらのルールは追加や消去が可能にように専用のエディタを開発している。

6. 話題ベース

6.1 話題ベースの構成

話題ベースには以下の情報が蓄積される。

- ・話題となっている時間、日、期間
- ・話題となっている人
- ・話題となっている行事
- ・ユーザーの要求事項
- ・話題の焦点

それぞれの情報は変化したもの以外は残される。話題の焦点はユーザーとの対話の履歴から話題になっているのは何かについて調べ、それを書込む。

6.2 話題ベースによる会話制御

話題ベースはユーザーの意図の履歴を短期記憶に蓄積し、以下の目的に使用する。

- ①入力フレームの不足情報の補完

②話題の変化に対応

ユーザーはある事について会話をしている時、一度言った事は省略することが多い。従って①には必須である。また話題が人から行事へ、行事から人へと変化した場合などに対応できるようにしている。

7. まとめ

本システムはPrologで約6.5k行（内日本語処理が約60%）の規模である。この結果、比較的簡単なシステム構成ではあるが種々の表現にも対応が可能で、応答速度も充分実用に耐える程度のもの（コンパイル版で1秒以内）が得られている。

しかしながら、常識の不足や、常識の適用順序の誤りなどのために我々の目から見て不自然に感ずる点も多々あり、まだまだ改善の余地がある。

このシステムの試作を通じて今後の課題をまとめると以下の通りである。

(1) Prologのメモリ領域の問題

現在はDEC10-Prologのversion3.4を使用しているが、すべての機能と知識ベース、データベースをメモリ上（約0.5MB）にロードしてプログラムを動作させるのは困難な状態になっている。従ってさらに大規模なPrologプログラムを動作させるためにはメモリ領域を大幅に拡張する必要がある。

(2) 日本語処理機能の拡充

現在の技術では、日本語処理機能の拡充は性能（特にレスポンスタイム）の改善とは相反する要求である。従って効率のよい高速な構文解析技術・意味解析技術の開発が必要である。

(3) 常識推論の強化

常識の範囲を限定することは困難だが、会話の不自然さをなくすためには、さらに幅広い常識を知識ベース化する必要がある。

謝辞

本研究の機会を与えて戴いた電電公社横須賀電気通信研究所知識ベース研究室寺島室長に深謝致します。またシステムの試作・試用に当ってご協力戴いた同研究室の各氏に深謝致します。

参考文献

- [1] 鈴木他：“日本語質問応答システムにおける知識表現と利用” 信学会 (AL82-70)
- [2] 東田他：“PROLOGによる知的秘書システムの試作” 第28回信学会全国大会 (6-157)
- [3] R.E.Fikes：“Odyssey:A Knowledge-Based Assistant” Artificial Intelligence 16(1981) pp.331-361.
- [4] I.P.Goldstein et.al.：“NUDGE, A KNOWLEDGE-BASED SCHEDULING PROGRAM” IJCAI'79 pp.257-263.
- [5] D.L.Bowen：“DECsystem-10 PROLOG USER'S MANUAL” Department of Artificial Intelligence, University of Edinburgh.
- [6] 長尾：“言語工学” 昭晃堂