知識工学と人工知能　41－4
（１９８５．７．４）

データベース構築支援エキスパートシステム
－ 基本構成と対話による論理設計支援 －

川口敦生[†]　溝口理一郎[††]　山口高平[††]　馬野元秀[†††]　磯本征雄[*]　角所収[††]
[†] 大阪大学工学部　[††] 大阪大学産業科学研究所
[†††] 大阪大学大型計算機センター　[*] 名古屋市立大学計算センター

１．はじめに

学術データベースは、一般に計算機の非専門家である各学問分野の研究者によって構築される。この際、データベース管理システム（DBMS）の活用が問題となる。彼らは計算機やＤＢＭＳについての知識が乏しいために、独力でデータベースの構築・管理を進めるには多くの障害が存在する。したがって学術データベースの発展には、構築・管理の容易なＤＢＭＳが提供されることが必要となる。

一方我々は、計算機の非専門家がデータベースの構築・管理を容易に行えるように計算機で支援することを考え、知的支援システムＫＤＢＭＳ[1]を開発してきた。これは、データベース構築に関する専門知識をデータベース化し、ＤＢＡ（database administrator）に作業の手順、助言を与えるものであった。

さて、知識工学の立場からは非専門家の支援を行う場合、データベースの専門家の振舞をシミュレートして支援するのが自然である。本稿では、こうした考えに基づくデータベース構築支援エキスパートシステムの構想を紹介し、非専門家と対話して論理設計を行うサブシステムについて述べる。

２．データベース構築支援エキスパートシステム

２．１　基本的な考え方

データベース構築・管理に際して、非専門家に最も望ましい環境は、身近にデータベースの専門家がいることであろう。彼は、専門家と相談しながら作業を進めることができる。さらに計算機の操作といった煩雑な作業の大半を専門家が代行してくれるために、構築依頼者は自身のデータの検討といった本質的な作業に専念できる。

しかし、上記のような環境はほぼ実現不可能である。そこで知識工学の手法を用いて専門家の振舞をシミュレートする、すなわち専門家の役割を計算機に肩代りさせることを考える。

本システムでは、次のような考え方を取っている。

・システム主導でユーザと対話しながら作業を進める。

・計算機の操作やＤＢＭＳの使用はすべてシステムが行う。

・対話には自然言語を用い、システムを使用するにあたってユーザがマニュアル等を参照する必要がないようにする。

２．２　システム概観

システムの概略を図１に示す。システムは一般的なエキスパートシステムにＤＢＭＳインタフェースを付加したものである。

知識ベースは、図に示すように４種類の知識から構成される。ａは、概念スキーマを作成するために必要な知識を獲

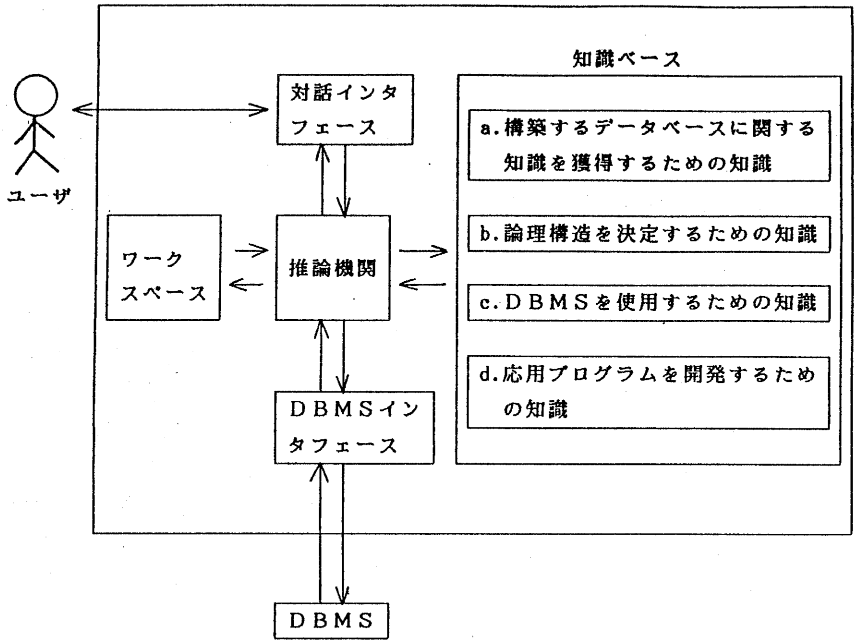

図１　データベース構築支援エキスパートシステムの概略

得するための知識である。ｂの知識は、対話を通して獲得された知識を用いて概念スキーマを作成するのに用いる。ｃはＤＢＭＳを使用する際に必要な知識で、用いるＤＢＭＳに固有の知識である。ｄは応用プログラムを開発するためのものである。

各知識に適した知識表現法が存在するので、推論機関もそれに応じて複数個存在する。後述するように概念スキーマの作成にあたっては、プロダクションルールおよびフレームによる知識表現が用いられる。

次章では、論理設計の支援について、特に知識獲得のための対話システムを中心に述べる。

３．対話にもとづく論理設計支援

３．１　目的と基本的考え方

一般にデータベースの論理設計では次のような手順を踏む。

(1) データベースに対する要求を把握する

(2) 要求を満たすデータモデルを構築する

ここで(2)においてデータに潜在する論理的な関係を陽に把握する必要がある。しかし、この作業は一般に初心者には困難であり、数度の試行錯誤を必要とする。

これまで我々は、論理設計を計算機を用いて支援することを考えてきた。そして

・検索要求文の中に、論理設計に必要かつ十分な情報が含まれている。

・特定のデータに関して、システムが持つべき知識を仮定しない。

という立場のもとに、あらかじめ用意された検索要求文に対して構文情報を主体とした解析を行うことによって論理構造を抽出するシステムを開発してきた。[2]

しかし、この方法には、次のような欠点があった。

・意味情報を利用しないために完全な論理構造を抽出することができない。

・必要十分な検索要求文をあらかじめ用意しておかねばならず、データベースを段階的に構築することができない。

さて、データベースの専門家が身近にいれば、初心者は相談しながら論理設計を行ない実際にデータベースを試作、使用しては改良を加えるということが可能になる。そこで専門家と初心者が対話しながら論理設計を行っている状況を考え、知識工学の手法を用いて専門家の振舞をシミュレートすることとした。

本システムでは次のような考え方をとっている。

・ユーザとの対話中に現れる名詞、動詞の意味および構文情報が論理設計に必要な情報を担っている。

・名詞の意味は構築するデータベースの分野に応じて変化することが予想されるので、あらかじめシステム内に辞書として用意しておくことはせずに、そのつどユーザに問合せ、知識ベースに蓄積していく。

・ユーザの要請に応じて随時得られた論理構造を持つデータベースをシステムが試作し、実際に使いながら論理設計を改良できるようにする。

ここで、名詞、動詞に注目するのは、論理設計のツールとしてE－Rモデル[3]を用いるためである。

３．２　論理設計支援システムの概要

論理設計支援システムの概要を図２に示す。この図は、図１に示した本エキスパートシステム全体の概略図中の知識ベースにおいて、ａとｂの知識に注目したものに対応する。

対話インタフェースは、日本語による入出力を行う。日本語構文解析部は文節ごとに区切って入力された日本語平叙文に対し、品詞の同定および文節間の係り受けの抽出を行う。[4] 結果は、プロダクションシステム（ＰＳ）のワーキングメモリー（ＷＭ）にアサートされる。日本語生成は、canned text によってい

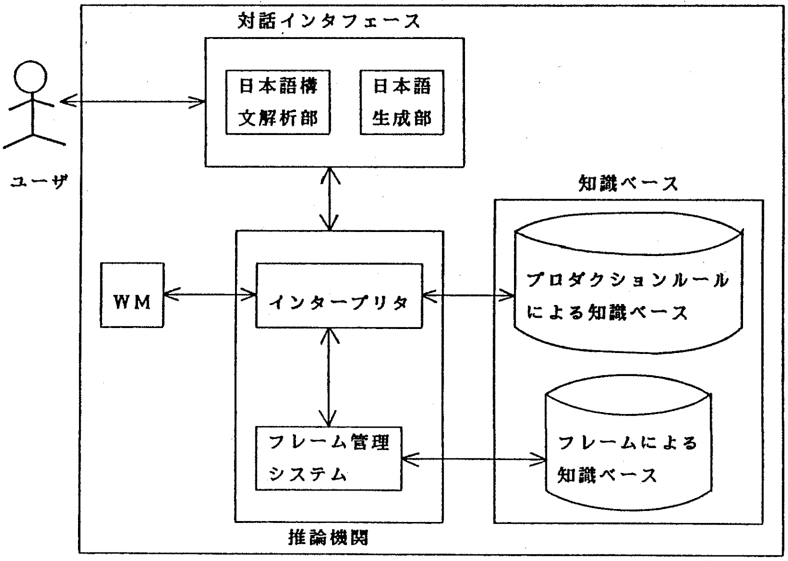

図２　論理設計支援システムの概要

る。

ＰＳは対話全体の制御を行う。ＷＭにアサートされたユーザの入力の解析結果に応じて対応するルールが対話の流れを制御する。

フレームシステム（ＦＳ）はユーザが構築しようとしているデータベースに関するモデルを構築する。フレームによる知識ベースに、データベースのデータ項目になると思われる諸概念に関する知識が格納されている。モデルの構築は、これらのあらかじめ用意された概念とユーザがデータベース化しようとしている概念との類似性を求めて行う。

３．３　実行例

システムの詳細な説明を行う前に、実行例を図３に示す。図では、ユーザ手持ちの文献に関するデータベースを構築しようとしている。（以下の説明で、番号は図３中の番号を付された対話を指す）

（１）まず、以後の対話を円滑に進めるために構築するデータベースの名前を聞く。

（２）対話のきっかけを掴むためにどのように使いたいかを聞く。

（３）文献についてシステムは知識を持っていないので、しばらく対話の焦点を文献に絞る。

（４）自分の知っている事柄のうちで最も近いものがどれかを聞く。それがわかれば、文献が持つ属性その他について質問を発っせられる。

（５）物理的な物について自分が持っている知識を述べる。また、文献を作る主体の存在が予見される。

（６）まず、作る主体と文献との関連を聞く。

（７）関連がわかったので、作る主体を聞く。

（８）著者と文献との対応を聞いておく。

（９）文献の他の属性をユーザが気がついているかもしれないので聞いておく。

(10) 文献には名前があって、著者と関連があることがわかった。著者についてもシステムは知識を持っていないので、次は著者に対話の焦点を絞る。

(11) 文献が人工の物だったので、著者は人間活動の主体であることがわかっている。

(12) 著者は人であることがわかった。システムがあらかじめ持っている知識から、どのような属性を持つか想像できるので提示してユーザに選んでもらう。

(13) ユーザが国籍を選んだので、著者には外国人が含まれることがわかる。あらかじめ持っている知識から名前の格納法を提案し、承諾を得る。

(14) 年齢と生年月日には、従属性が存在する。両方を選んだときは、データベースには生年月日を格納し、年齢は計算で求める。

(15) システムが人について知っている属性以外の属性が著者に存在するかどうか聞く。

(16) 文献と著者について必要なことがわかった。対話をさらに進めるために他にユーザがしたいことを聞く。

３．４　対話の制御

全体の対話の流れは、ＰＳが制御する。一つ一つの話題についての対話はフレームの attached procedure によって行なわれる。

ＰＳのルールには、大きく別けて次のような種類が存在する。

構築するデータベースの名前を入力してください。 ・・・・1
　　ｐａｐｅｒ＿ｄｂ

ｐａｐｅｒ＿ｄｂで何をしたいのですか。
　　文献の　名前を　データベースに　しまっておきたい ・・・・2

「文献」について教えてください。 ・・・・3
次の概念の中から、文献に最も近いものを選んでください。 ・・・・4

1:抽象的関係	6:人間活動の主体	12:自然物および自然現象
2:時間	7:個人	13:刺激
3:空間、場所	8:組織	14:自然、物体、物質
4:量	9:人工の物	15:生物
5:単位	10:物理的な物	16:体
	11:抽象的な物	17:その他

どれですか。
　　１０
通常、物理的な物には名前とそれを作る主体があります。 ・・・・5
「文献を作る」ことを何と言いますか。 ・・・・6
　　書く
文献を書く「主体」を何と言いますか。 ・・・・7
　　著者
一つの文献に著者は、１　一つだけ　２　二つ以上　のどちらですか。 ・・・・8
　　２
文献に必要な他の属性があれば入力してください。 ・・・・9
　　ありません。

「著者」について教えてください。 ・・・・10
次の概念の中から、文献に最も近いものを選んでください。 ・・・・11

1:人間活動の主体
　2:個人
　3:組織

どれですか。
　　２
通常、個人には次のような属性が存在します。著者に必要な属性を選んでください。 ・・・・12
１名前　２性別　３国籍　４住所　５年齢または生年月日　６所属
どれですか。
　　１　３　５
著者には日本人だけでなく外国人もいるのですね。 ・・・・13
著者の名前として、first name, second name, last name をｐａｐｅｒ＿ｄｂに
格納します。よろしいですか。
　　はい
年齢または生年月日が必要なのですね。 ・・・・14
１年齢　２生年月日　３両方のどれをｐａｐｅｒ＿ｄｂに格納しますか。
　　１
著者に必要な他の属性があれば入力してください。 ・・・・15
　　ない

他にしたいことがあれば、入力してください。 ・・・・16
・
・
・

図３　論理設計支援システムの実行例

・ユーザに入力を促すルール

・入力文中に出現した単語を話題として対話を続けるためのルール

・不要になったWM要素を削除するためのルール

話題に関するルールは、以下の原則に従って記述されている。

・FSの知識ベースにない名詞、動詞に注目する。

・係り受け関係によって最も情報が多い単語に注目する。

・FSの知識ベースに既にある単語と係り受け関係がある単語に注目する。

FSの知識ベースにない名詞が入力文に現れたときの対話は次のような処理によって行う。

(1) FSの知識ベースにある概念をすべて表示し、最も近い概念を選ばせる。

(2) 選ばれた概念を上位概念とする新しい名詞のフレームをFSの知識ベースにアサートする。(このとき、同時にその名詞のメタフレームが生成される。)

(3) 新しい名詞のフレームに必要なスロットを作るために、メタフレームのスロットの値($value)を読みだす。これによって $if_needed attached procedure が起動され、ユーザに必要な質問を発っしながら適切なスロットを生成する。

上記の処理が行なわれる様子を、図3の実行例について以下に示す。図4は使用されたプロダクションルールの例である。また、図5にメタフレームの例を示す。図中に付された番号は実行例(図3)の番号に対応する処理を示している。

まず、システム起動時に自動的にルール13が発火し、対話1が行なわれる。その後ルール12によって対話2が行われ、入力文を得る。この文に対してはルール70がマッチし、話題を「文献」に絞る。

ルール70によって文献フレームが作られる。そして文献メタフレームのすべてのスロットの値を読みだす。実際には、文献メタフレームには何のスロット

```
ルール13:
    IF      WMに何もない。
    THEN    構築するデータベースの名前を聞き、                 ---→  1
            それをWMにアサートし、
            WMに「一回目の質問」とアサートせよ。
ルール12:
    IF      「一回目の質問」である。
    THEN    「一回目の質問」をWMから取り去り、
            何をしたいかを聞け。                               ---→  2
ルール70:
    IF      WMに名詞があり、
            その名詞のフレームがフレーム知識ベースにない。
    THEN    その名詞について教えてくださいと出力し、           ---→  3
            最も、近い概念を聞き、                             ---→  4
            その名詞のフレームをフレーム知識ベースに作り、
            そのメタフレームのスロットを全て読みだせ。
ルール75:
    IF      WMに名詞とそれに近い概念が組で存在し、
            その名詞のフレームがフレーム知識ベースにない。
    THEN    その名詞について教えてくださいと出力し、           ---→  10
            近い概念をもとにさらに近い概念を聞き、             ---→  11
            その名詞のフレームをフレーム知識ベースに作り、
            そのメタフレームのスロットを全て読みだせ。
```

図4 プロダクションルールの例

も存在しないため、継承によって物理的な物メタフレームが呼び出される。

物理的な物メタフレームには、図5に示すような attached procedure が存在し、対応する番号の対話を行う。この際、名詞「著者」が対話中に現れる。

文献に関する対話が一段落すると、著者に対話の焦点を移して同様な処理が行なわれる（ルール75）。話題がなくなれば再びユーザがしたいことを聞き、対話を続ける。

図6は、対話1～15を通して獲得された知識である。文献フレームは、「文献」という主体が名前という属性を持ち、詳細は普通の名前（normal_name）フレームに記述されていることを示している。さらに、動詞「書く」を関連とする主体「著者」が存在し、「文献」と1：nの対応関係があることも示している。

また、著者フレームは、主体「著者」が名前、国籍および年齢または生年月日という属性を持ち、それらに関する知識がそれぞれ、外国人の名前（name_of_foreigner）、国籍（nationality）および年齢（age）フレームに記述されていることを示している。

これらの事柄は、図7のE－Rモデル表現に対応する。

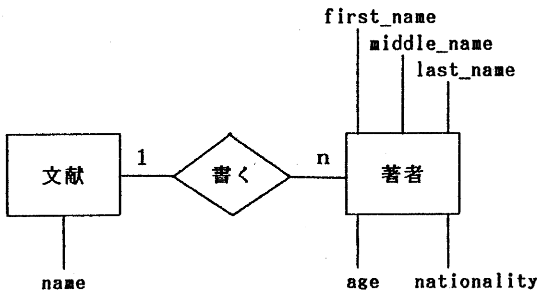

図7　図3の対話例から得られたE－Rモデル

```
物理的な物
    一般的な属性
        $if_needed
            print_general_attribute  ーー→5
        $data_for_if_needed
           [name, maker]
    作ること
        $if_needed
            ask_verb_means_make      ーー→6
    作る主体
        $if_needed
            ask_noun_means_maker,    ーー→7
            ask_number_of_maker      ーー→8
    他の属性
        $if_needed
            ask_other_attribute      ーー→9

個人
    一般的な属性
        $if_needed
            print_general_attribute
                 _and_let_select,    ーー→12
            check_nationality,       ーー→13
            check_age_or_birth       ーー→14
        $data_for_if_needed
           [name, sex, nationality,
            address, age_or_birth,
            belong_to]
    他の属性
        $if_needed
            ask_other_attribute      ーー→15
```

図5　メタフレームの例

```
          文献
              上位概念
                  $value
4ーー→                physical_object
              下位概念
                  $value
                      []
              名前
                  $value
5ーー→                kind: normal_name
7ーー→        著者
                  $value
6ーー→                relationship: 書く
8ーー→                number: n

          著者
              上位概念
                  $value
11ーー→               individual
              下位概念
                  $value
                      []
12ーー→       名前
                  $value
13ーー→               kind: name_of_foreigner
12ーー→       国籍
                  $value
                      kind: nationality
12ーー→       年齢または生年月日
                  $value
14ーー→               kind: age
```

図6　獲得された知識

３．５　検討

論理設計のために構築依頼者と対話しているデータベースの専門家の振舞を考えた場合、次のような点に注意しているものと思われる。

(1) 自明の事柄は、専門家から依頼者に積極的に提示する。

(2) 隠れがちな論理的関係を依頼者が気付けるように、よいきっかけをあたえる。

(3) 今どのような論理構造がえられているかを常に依頼者に意識させ、考えさせる。

上記の観点から、本システムでは以下の事柄を実現した。

(1) 一般的な諸概念について知識をあらかじめ持たせる。（対話５あるいは１２）

(2) 忘れがちな事柄をシステムが知識として持っており、ユーザに示唆する。（対話１３）

(3) システムが構築している対象モデルを、Ｅ－Ｒモデル表現で常にユーザに提示する。

今後、さらにシステムを有用にするために、次のような事柄を実現する予定である。

(1) 対象モデルに基づいてデータベースを構築した場合に、どのような検索ができるかを随時ユーザに提示する。

(2) ユーザの求めに応じてデータベースを試作し、使用後に論理設計をやり直す。

また現在は、関数従属性等の正規化を行う際に必要な知識を対話から得ていない。それらの概念が非専門家には難解で不必要なものと考えているためである。正規化に関しては、試作したデータベースに入力されるデータから、必要な情報を抽出することを考えている。

試作データベースでは、上述の情報の抽出および短期間での試作が要求される。そこで、Prolog データベース上でＤＢＭＳをシミュレートするプログラムを Prolog で作成中である。

４．おわりに

知識工学の手法を用いて、データベースの構築を支援するシステムとその論理設計支援サブシステムについて述べた。

論理設計支援サブシステムは、文献データベースの構築に適用して評価中である。今後、より論理構造の複雑な分野に応用し、論理設計を行う際に必要な知識、対話等について詳細に検討を加えていく予定である。

現在、関係型のＤＢＭＳであるＤＧ／ＳＱＬを対象とし、ＭＶ／８０００Ⅱ上のＭＶ－Ｐｒｏｌｏｇを用いてインプリメントを行っている。稼働しているのは論理設計支援サブシステムのみであるが、他のサブシステムも早急にインプリメントしシステム全体の評価を行っていきたいと考えている。

参考文献：

[1] 磯本他:"データベース構築・管理のための知的支援システム -knowledge based DBMS(KDBMS)-"「アドバンスト・データベースシステム」シンポジウム，情報処理学会，昭和５７年１２月．

[2] 小林他："データベース論理設計支援エキスパートシステム"，知識工学と人工知能研究会資料33-2，昭和５９年１月．

[3] Chen,P.P.:" The Entity-Relationship Model : Toward a Unified view of Data", ACM TODS 1, pp.9-36 (1976).

[4] 吉田将："二文節間の係り受けを基礎とした日本語文の構文分析"，信学論誌，vol.55-D No.4 (1972).