知識工学と人工知能 33－2
（1984. 1. 26）

データベース論理設計支援エキスパートシステム

小林 仁[+] 溝口 理一郎[+] 磯本 征雄[#] 豊田 順一[+] 角所 収[+]
（+ 大阪大学産業科学研究所） （# 大阪大学大型計算機センター）

1。まえがき

我々は、一般の研究者が容易にデータベースを構築・管理できるための知的支援システムＫＤＢＭＳ(Knowledge-Based DBMS)[1][2]を開発してきた。ＫＤＢＭＳの開発の背景には、学術データベースの構築・管理にとっては、そこで取り扱われる情報を熟知した専門家が直接関与するのが最良であるが、彼らはデータベース管理システム（ＤＢＭＳ）や計算機システムの活用に関しては非専門家であり、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あるという事実がある。ＫＤＢＭＳは、今までのデータベース構築・管理の支援を通して得られた経験的知識をデータベース化して、計算機ないしＤＢＭＳ自体にＤＢＭＳ利用支援の面倒を見させようというものであり、最終的にはマニュアルレスシステムの形態を目指している。

データベース構築において、初心者が遭遇する困難な点は次の５つにあると思われる。

１）ジョブ制御言語の使い方
２）全体の処理の流れの把握
３）誤りからの回復
４）論理設計
５）手持ちのデータのデータベースへの格納

これら５つの項目全てを支援することが、我々の最終目的であることは言うまでもない。１）～３）に関してはプロトタイプシステムが稼働している状況にあり[1][2]、現在、４）と５）に関する検討を行っている。本稿では、４）の論理設計を支援するサブシステムについて、その概要を述べる。

2。基本的な考え方

データベースの概念スキーマを作成するにあたり、データベースに関して初心者であれば、まずマニュアルを読むことから始めなくてはならず、概念スキーマとは何かということを理解するまでに、多大な負担を強いられる。その上、さらに実際に手持ちのデータに潜在する論理的な関係を陽に把握して、データベースにおける表現に適したように再構成する事は、非常に困難な作業になる。大規模な商用データベースシステム等を構築する場合には、データベースの専門家と綿密な検討を行う必要があるが、学術データベースなどの場合にはできるだけ容易に設計できることが望ましい。我々の論理設計支援システムは、このようなデータベース設計に不慣れな初心者を支援することを目的として開発されている。[3] このシステムは、データベースを構築しようとする、本システムの利用者から得た検索要求文の構文情報をもとにして、会話的にデータベースの論理構造の設計を進めていくものである。

我々のシステムの基本的な考え方は、次に示す通りである。

（１）データベースへの検索要求文の中には、論理設計に必要かつ十分な情報が含まれている。

データベースの論理構造は、データが持つセマンティクスに係ることであるので、完全な自動抽出をすることはかなり困難なことと思われる。従来この問題に関しては、実際のデータ（値）を用いて、種々の解析技法を用いて論理設計の専門家が手探りで行うというアプローチがとられていた。本稿では、データの論理構造を知る手掛かりとして、自然言語によるデータベースへの検索要求文を用いることにする。一般に検索要求文はかなり制限された形をしており、各検索文に固有の論理構造を持っている。従って、十分な数の検索要求文が与えられれば、それらの検索要求に応じる為に必要なデータベースの論理構造を推定することが可能であると考える。

（２）特定のデータに関する、システムが持つべき知識は原則として仮定しない。

正統的なアプローチとして、データの値に関する情報を用いてそれらの間の関係の同定、バリューアブストラクション（値の抽象化）による属性の抽出等を行うことにより、論理構造を「学習」する方法が考えられる。このアプローチは学問的に見ても帰納的推論との関連が示唆されて、興味深いところがある。しかしながら、たとえこの方法が理論的にある程度解決されたとしても、次の理由によりそれは現実的なシステムに組み込むことはできない。一般には事前にどのようなデータがくるか判らない為、多種大量のデータ（値）に関する情報とそれらを操作する知識を蓄積していなければならないが、これは事実上不可能である。従って、このアプローチを採用するシステムでは、利用者は各自が所有するデータのセマンティクスを（各値の関係をシステムが要求する形に）整理する操作が必要となる為利用者

の負担は大きく、悪くするとデータベースの論理設計と大差ないことにもなりかねない。

（３）構文情報を主体とした解析を行う。

上述の理由により、名詞以外の単語に関する辞書を持ち、未知の単語は全て名詞として扱い、構文情報を中心に処理を行うという立場をとる。このことにより、上述の、システムが持つべき辞書の複雑さの問題は解消される。必要な意味情報はできるだけ利用者との対話により得ることとし、利用者の負担を最小限に留めるように努める。

利用者からみた本システムの特徴として、次の３点が挙げられる。

（１）本システムの利用者は、事前に検索要求文を用意するだけでよい。

データベースの開発者にとってデータベースの論理設計は困難であっても、開発しようとしているデータベースの使い方に関する考えは持っているものと思われる。従って、ある程度制限された自然言語で検索要求の例を列挙することを要求することは開発者にとってそれほど大きな負担にはならない。

（２）会話的に設計を進めて行く。

利用者はシステムからの質問に答えるだけでよく、ほとんどがＹＥＳ／ＮＯの選択を中心とした即答できる質問であるので、利用者の負担は軽い。

（３）システムの利用者は、データベースに関する専門的知識を一切必要としない。

以上述べたように、本システムでは利用者の作業は完全に受動的であり、論理構造がほぼ自動的に生成されるので、データベースの初心者であっても大した負担を感じずに容易にデータベースの論理設計が行える。

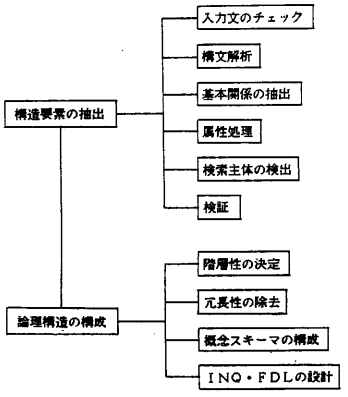

図１　処理の概要

３．システムの概要

本システムの処理手順の概略を図１に示す。ＤＢＭＳとして、日本電気（株）のＩＮＱ[4]を想定しているので、最終的には概念スキーマを、ＩＮＱのファイル記述言語ＦＤＬを用いて設計することを目的としている。しかしながら、概念スキーマの設計のツールとしてＥＲモデル[5]を用いており、基本的には他の関係データベースシステムにも対応できるように設計されている。

入力となる検索要求文は、複雑な文の出現の防止と解析の容易さの為、英語とし、さらにFindで始まる命令文に限定する。検索要求文のいくつかの例を以下に示す。

検索要求文の例

(a1) Find the names of employees in the toy department.

(a2) Find the items sold by departments on the second floor.

(a3) Find the avearge salary of the employees in the shoe department.

(a4) Find the departments that sell pens.

(a5) Find the companies that supply all the items of type A to all the departments on the second floor.

処理は、検索要求文からの構造要素の抽出と、得られた情報をもとに概念スキーマを構成する論理構造の構成に大別される。

本システムは拡張PROLOGである ShapeUp [6]でインプリメントされている。

尚、以下示されている実行例においてプロンプト？で始まる行は、利用者の入力を示す。

４．構造要素の抽出

入力された検索要求文から、論理構造を構成するための基本的な情報の抽出を行う。

４．１　入力文のチェック

検索要求文に、等位接続詞(and,or)が含まれていると、構文解析が非常に困難になるので、等位接続詞に対して次に示す前処理を行う。もし等位接続詞が存在すれば、検索要求文を意味がほぼ等価な２文に分割する。その結果を利用者に示し、了解が求められた場合には、改めてそれらを入力とする。もし、拒絶された場合は、元の検索要求文が破棄される。

２文に分割する方法は、まず等位接続詞の前までを第１文とし、等位接続詞の後ろの単語と同じ品詞の

単語をその接続詞の前の文中にさがし、みつけられた単語以前の文と接続詞以降の文とを結合して第2文とする。

等位接続詞の解析は自然言語処理の分野においても未解決の問題であるので、ここではこのような単純な処理を行った。この処理は、その単純さにもかかわらず適用範囲が広いことが確かめられている。論理構造を構成するために必要な検索要求文において、等位接続詞は不可欠なものではないので、利用者には、あまり使わないように指示することにする。実際、システムが前処理した結果が拒絶された場合には、その旨のメッセージを出すことにしている。

検索要求文分割の例

```
>:-INPUT.
?FIND THE ITEMS SUPPLIED BY LEVI
AND SOLD IN THE TOY DEPARTMENT.

CAN THIS SENTENCE BE DIVIDED INTO
NEXT TWO ?

FIND THE ITEMS SUPPLIED BY LEVI.
FIND THE ITEMS SOLD IN THE
TOY DEPARTMENT.

? YES.
```

4．2　構文解析

入力された検索要求文を構文解析し、構文木の生成を行う。構文解析のためのパーサーは、PROLOGのＤＣＧ[7]と同様であり、文頭から文脈自由文法の規則を適用するものである。但し、データのセマンティクスに関する情報、即ち名詞に関する辞書は持ち合わせていないので、未知の単語が現われるとそれを名詞と判断して解析を進める。関係詞又は代名詞の同定は、本パーサーでは行えないので、利用者に問い合わすことにより解決する。

また、生成された構文木には、以後の処理で必要な情報しか残していない。必要な情報とは、(1) 名詞および名詞と名詞を接続している品詞、(2) 名詞の位置に関する情報である。

（１）生成された構文木の例

```
>:-PARSER.
?FIND THE NAMES OF EMPLOYEES IN
THE TOY DEPARTMENT.

VP(V(FIND),NP(N([ITEMS],PP(P(OF),
NP(N([EMPLOYEES]),PP(P(IN),
NP1(N([TOY,DEPARTMENT])))))))).

>:-PARSER.
?FIND THE ITEMS SOLD BY NO
DEPARTMENT ON THE SECOND FLOOR.

VP(V(FIND),NP(N([ITEMS]),AP(V_PP(SOLD),
PP(P(BY),NP(N([NO,DEPARTMENT]),
PP(P(ON),NP1(N([SECOND,FLOOR])))))))).
```

（２）関係詞の同定の例

```
(A)
?FIND THE DEPARTMENTS THAT SELL PENS.

DOES THIS [THAT] DESIGNATE [DEPARTMENTS] ?

? YES.

(B)
?FIND THE ITEMS OF TYPE A WHICH ARE SOLD
IN THE TOY DEPARTMENT.

DOES THIS [WHICH] DESIGNATE [TYPE,A] ?

? NO.

WHAT DOES IT DESIGNATE ?

? ITEMS.
```

4．3　基本関係の抽出

名詞に関する辞書を持っていないので、意味的な解析は行えない。そこで、最も基本的な論理関係を示す係り受け関係に注目する。構文解析によって得られた構文木から、前置詞、関係詞、動詞等をキーワードにして、それと構文的に係っている名詞間の関係を抽出する。

抽出された基本関係の例（(a1),(a2)より）

```
RELATION([OF,[NAMES],[EMPLOYEES]]).
RELATION([IN,[EMPLOYEES],[TOY,DEPARTMENT]]).
RELATION([SELL,[DEPARTMENTS],[ITEMS]]).
RELATION([ON,[DEPARTMENTS],[SECOND,FLOOR]]).
```

RELATION の第１引数が抽出された基本関係の関係名であり、第２引数以降はその引数である。

ここでの処理は基本関係が互いに独立という考えに立っているが、それが誤りである場合がある。

次の例を考える。

(1) ~ type of items sold by departments ~

(2) ~ volume of items sold by departments ~

この例で得られる基本関係は、次の通りである。

(1)より

```
RELATION([OF,[TYPE],[ITEMS]]).
RELATION([SELL,[DEPARTMENTS],[ITEMS]]).
```

(2)より

```
RELATION([OF,[VOLUME],[ITEMS]]).
RELATION([SELL,[DEPARTMENTS],[ITEMS]]).
```

基本関係が独立という立場で、(1) からは正しい結果が得られたが、(2) から得られた基本関係は正しくない。つまり、(1) における TYPE は ITEM だけに依存する属性であるが、(2) における VOLUME は ITEM にだけ依存するのではなく SELL にも依存する。従って、(2) から得られるべき基本関係は、

RELATION([SELL,[DEPARTMENTS],
[ITEMS],[VOLUME]]).

でなければならない。このように、基本関係が独立という仮定の誤りを補うために、いくつかのパターンをもつことにより対処することにしている。例えば、 *V は動詞、*P は前置詞、*A,*B,*C は名詞とするとき、

RELATION([*P,*A,*B])
RELATION([*V,*C,*B]).

という基本関係が１つの検索文から得られたならば、 *A が *B と *V の両方に関係しているかを利用者に問い合わす。

もし、その答えが YES であれば、その２つの基本関係を

RELATION([*V,*C,*B,*A]).

に統合する。

NO であれば、基本関係は独立として扱う。

この処理はＥＲモデルにおける、関連が属性を持つ場合の処理に対応している。

４．４　属性処理

検索要求文に現われている名詞は、属性名或は属性値のいずれかである。従って、構文解析によって名詞と判断されたものから、修飾関係を利用して属性名の抽出を行う。

ここで名詞と言っているものの中には、通常複数の単語（例えば toy department ）からできているものが含まれている。一般に２つの名詞で構成される名詞句では、前の単語が後ろの単語を意味的に修飾している。従って、最後尾の単語が属性名を示すと考えられ、それを語尾処理* することによって属性名を得る。しかしながら、 type A のように前者の単語の方が属性名を示す場合がある。そのような名詞（ type,size,length等のいわゆる属性名詞 ）は、数も限られているので、予め辞書に持っておくことにより判断する。

また、単語１語により名詞と抽出されている場合には、それ自身が前置詞等により後ろから構文的に修飾されているかどうかで判断する。修飾を受けていなければ属性値とし、修飾を受けておれば、語尾処理をしたのち属性名とする。

属性名抽出の例（ _が属性名 ）

(1)　toy department
(2)　type A
(3)　　that sell pens.
　　that sell items of type A.

４．５　検索主体の検出

検索主体とは、種々の属性を持つ、現実世界に存在する実体と考えられ、検索要求文では、Findの実質的な目的語に相当する。従って、その検索主体の検出は次のように行う。

次の２つの検索要求文を考える。

(1) Find N1 ～
(2) Find N2 of N1 ～

通常は N1 を検索主体とする。(1) の場合は明らかであろう。(2) の場合には、次の2つの場合があり得る。（下線部が検索主体）

(2-1) Find the names of employees who ～
(2-2) Find the items of type B which ～

即ち、N1 に TYPE 等の属性名詞が含まれていなければ N1 を、含まれていれば、N2 を検索主体とする。

４．６　検証

以上の処理は、個々の検索要求文について行われるが、全体を通してみてさらに正確な情報の獲得を行う。

（１）属性処理について

名詞が２つの単語から構成されている時、属性名詞が含まれる場合を除き最後尾の単語が属性名を示すとしたが、その処理では不都合が生じる場合が存在する。特に、学術名に多いと考えられる。例えば atom C, vowel A では、 C や A が属性名となってしまう。しかしながら、このような問題は、次のような場合に限り自動的に検出、修正できる。

○２つの単語からなる各々の名詞の中で、前の方の単語が等しく、後ろの単語が異なる名詞が存在すれば、その前の方の単語を属性名とし、後ろの単語で得られた属性名を削除する。

例：

VOWEL A, VOWEL E ⟶ VOWEL
ATOM C, ATOM H ⟶ ATOM

また、ある属性がある関係においては属性値でしか現われていないことが考えられる。従って、そのような場合に対処するために属性値が含まれる属性名を推定する必要がある。本システムでは、次の場合に限り推定を行うことができる。

○注目する属性値を含む基本関係と同じキーワードをもつ基本関係の対応する要素位置にある属性名に、その属性値は属するものと推定する。但し、

---

* 複数形を単数形に変換している。

このキーワードは動詞のときだけである。そして、推定した結果を利用者にnoと答えられたとき或は推定できなかったときは、その属性名を利用者に質問する。以下に例を示す。

属性名推定の例

（属性処理された基本関係　B_RELATION）
（属性名　ATTRIBUTE）
（属性値　ATTRIBUTE_V）

```
B_RELATION([SELL,DEPARTMENT,PEN]).    ①
B_RELATION([SELL,DEPARTMENT,PENCIL]).②
B_RELATION([SELL,DEPARTMENT,ITEM]).   ③
ATTRIBUTE([ITEM]).
ATTRIBUTE_V([PEN]).
ATTRIBUTE_V([PENCIL]).

IS [PEN] A VALUE OF [ITEM] ?

? YES.

IS [PENCIL] A VALUE OF [ITEM] ?

? YES.
```

ここで、3番目の B_RELATION がない、或は、ITEM が属性名と判っていない、或は上の質問で NO と答えられたする。この場合、上記の PEN, PENCIL のような同じ構造(①,②)をもつものをグループとして集め、そのグループの名詞、この場合では PEN と PENCIL、が同一の属性名を持つかどうかを利用者に問い合わす。もし持つならば、今までに得られている属性名の中から対応するものを選択してもらう。そうでなければ、利用者にその属性名を問い合わす。グループをなさなかった時には、個々の属性値対して同様な操作をする。

さらに、以上のすべての結果を利用者に示し、確認を求める。

```
>:-CHECK.

*** ATTRIBUTE NAME ***
      []
      [NAME]
      [EMPLOYEE]
      [DEPARTMENT]
      [ITEM]
      [FLOOR]
      [SALARY]
      [COMPANY]
      [TYPE]

*** ATTRIBUTE VALUE ***
      []
      [PENS]

**** RELATION ***
OF   <----   [NAME,EMPLOYEE]
IN   <----   [EMPLOYEE,DEPARTMENT]
SELL   <----   [DEPARTMENT,ITEM]
ON   <----   [DEPARTMENT,SALARY]
OF   <----   [SALARY,EMPLOYEE]
SUPPLY   <----   [COMPANY,DEPARTMENT,ITEM]

ATTRIBUTE NAME OK ?
```

5．論理構造の構成

上述の構造要素の抽出によって得られた情報をもとにＥＲモデルによる表現を行った後に、概念スキーマを形成する。

本システムでは、ＥＲモデルの関連が更に階層と関連に区別されている。ＩＮＱのＦＤＬを用いて概念スキーマを設計することを考慮すると、２項関係においてはいかなる対応関係があるかが重要であり、２項関係だけを示す前置詞、関係詞については階層として対応関係を明らかにすることにしている。

5．1　階層性の決定

得られた基本関係を関連(RELA)に属するものと、階層(HIER)に属するものに区別する。関連に属するものは、基本関係のキーワードが動詞である場合であり、階層に属するものはそのキーワードが前置詞、関係詞、一部の動詞の場合である。さらに、階層に属するものに関してその対応関係を明らかにする。

関連 ――― SUPPLY,SELL,・・・・
階層 ――― IN,ON,OF,・・・・
　　WHOSE,・・・・
　　HAVE,・・・・

階層に属する基本関係は全て２項関係であり、実際に設計される概念スキーマでは、その関係は一方が他方の属性（ＩＮＱでは項目）として含まれるように表現される。

その一方が主体を示し、他方が属性である２項関係の場合には、この属性は必ずその主体に含まれるので、主体が決まればその属性がユニークに決まるかどうかだけを、利用者に問い合わせて明らかにする（主体：属性　１：１，１：Ｎ）。また、両者が主体である場合には、次の４つの内のどの関係が存在するか明らかにする。

主体：主体　１：１
　　１：Ｎ
　　Ｎ：１
　　Ｎ：Ｍ

ただし、利用者に対する負担を軽減するために、利用者への質問の数を極力減らすことにしている。そのため、例えば IN などのようにその対応関係が明らかなもの（上位、下位概念の対応）に関しては、主体同志の包含関係は明らかであり、概念スキーマの表現のためには１：１か１：Ｎかの情報は必要な

いので、そのことについての質問はしない。

階層性決定の例：

今、次の基本関係（＿は検索主体を示す）が得られているとする。

```
B_RELATION([SELL,[DEPARTMENT],[ITEM]]).
B_RELATION([OF,[NAME],[EMPLOYEE]]).
B_RELATION([OF,[DEPARTMENT],[EMPLOYEE]]).
```

この時、次の３つの質問が発せられる。

```
IS [NAME] UNIQUELY DETERMINED ACCORDING
TO THE SELECTION OF [EMPLOYEE] ?

? YES.

IS [DEPARTMENT] UNIQUELY DETERMINED ACCORDING
TO THE SELECTION OF [EMPLOYEE] ?

? YES.

IS [EMPLOYEE] UNIQUELY DETERMINED ACCORDING
TO THE SELECTION OF [DEPARTMENT] ?

? NO.
```

１番目の質問は、主体と属性の間の階層関係を聞いたものであり、EMPLOYEE:NAME が 1:N でないことが分かった。２，３番目の質問は、主体同志の階層関係を聞いたものであるから、２回必要である。この結果、

DEPARTMENT:EMPLOYEE=1:N

であることが分かった。

得られた結果は次のようになる。

```
RELA([SELL,DEPARTMENT,ITEM]).
HIER([EMPLOYEE,NAME],1).
HIER([DEPARTMENT,EMPLOYEE],2).
```

但し、RELAの第１引数は関連名を示す。HIERの第２引数は対応関係を示し、下記の意味を持っている。

1:1=1，1:N=2，N:1=3，N:M=4

５．２　冗長性の除去

以上で得られた関係には、相互に冗長性が存在する可能性がある。次のような場合には、その検出および除去ができる。

（１）関連と階層の間の冗長性

・階層で得られた基本関係の２項ともが、関連とされた基本関係に含まれている場合がある。この場合は容易に検出でき、階層として得られた関係を除く。

例：

```
RELA([SELL,DEPARTMENT,ITEM]).
HIER([IN,ITEM,DEPARTMENT],1).
```

（２）関連と関連の間の冗長性

・関連を示すキーワードが一意に命名されていない場合がある。この場合は、異なる関連名を持ち同じ属性名の集合を持つ関連を検出することにより解決する。

例：

```
RELA([SUPPLY,COMPANY,DEPARTMENT,ITEM]).
RELA([RECEIVE,DEPARTMENT,ITEM,COMPANY]).
```

この場合、２つの関係が同じ意味かどうかを質問し、同じであれば、いずれか一方を除去し、異なっていれば、両方とも残す。

・主体が一意に命名されていない場合がある。この場合は、同じ関連名を持ち、その属性も１個所だけ名前が異なるが他の位置に関しては同じ名前である関連を複数検出することにより解決する。

例：

```
RELA([SUPPLY,SUPPLIER,ITEM]).
RELA([SUPPLY,COMPANY,ITEM]).
```

次のような質問により、了解が得られれば、一方を除去する。

```
IS [SUPPLIER] EQUAL TO [COMPANY] ?
```

・同じ関連に含まれるべき属性が異なる関連に別れてしまっている場合がある。

例：

```
RELA([SUPPLY,COMPANY,DEPARTMENT]).
RELA([SUPPLY,COMPANY,ITEM]).
```

次のような質問の後、マージを行う。

```
IS THE FOLLOWING RELATION CORRECT ?
RELA([SUPPLY,COMPANY,DEPARTMENT,ITEM]).
```

（３）階層と階層の間の冗長性

・異なる主体が同じ属性を持つことが誤りである場合がある。

例：

```
HIER([EMPLOYEE,DEPARTMENT],3).
HIER([EMPLOYEE,SALARY],1).
HIER([DEPARTMRNT,SARARY],1).
```

この例の場合、主体 EMPLOYEE と DEPARTMENT が、共に属性 SALARY をもつという関係が抽出された。主体同志の関係は，EMPLOYEE:DEPARTMENT=N:1 であることを示している。この DEPARTMENT と SALARY の基本関係は、

～departments having average salary ～

のような検索要求文から抽出されるのであるが，「average」という単語が、DEPARTMENT と EMPLOYEE との階層性から暗に DEPARTMENT に含まれる EMPLOYEE に用いる平均操作を示しているため、DEPARTMENT と SALARY が直接の階層関係を持つことが誤りであることが検出される。このように、平均、総和を示す average,total等に注目することにより誤りを訂正することができる。このような単語がなかったならば、次のような質問をした後、除去を行う。

IS THERE A DIRECT RELETION BETWEEN
[SALARY] AND [DEPARTMENT] ?

5．3　概念スキーマの構成

上述の処理により得られた結果からＥＲモデル表現は容易に構成できる。図2に4．2までの処理で得られた結果のＥＲモデル表現を示す。

本稿では、検索文に反映しているデータの論理構造を素直に表現することを目的としており、関係モデルにおける正規化[8]の操作は行わない。

概念スキーマの設計は、関連および検索主体を中心に行う。具体的には、用いられるＤＢＭＳ、我々の場合にはＩＮＱの特性を考慮した設計を行う。ＩＮＱでは、階層的に定義された複数のファイル（関係）を関係データベース的に用いることができるので、この特性を有効に利用する。

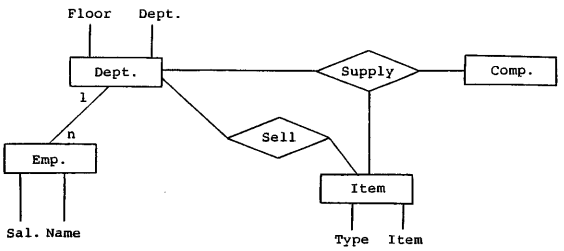

図2　ＥＲモデル図表現
（検索文(a1)～(a5)）

ａ）関連(RELA)についてはそのまま用いる。即ち、関連名をファイル名とし、その属性を単一項目として加える。

例えば、

RELA([SUPPLY,COMPANY,DEPARTMENT,ITEM]).

に対して

SUPPLY(COMPANY,DEPARTMENT,ITEM).

のような概念スキーマを作成する。

階層については、主体ごとにファイル（関係）を作成し、以下の操作に従って属性（検索項目）を追加していく。

ｂ）検索主体と属性による階層については、その属性値がユニークに決まる場合には、その属性を主体に単一項目として付加する。ユニークに決まらない場合には、不定繰り返し項目として付加する。

例えば、EMPLOYEEが検索主体で他が属性のとき

HIER([EMPLOYEE,NAME],1).
HIER([EMPLOYEE,SALAEY],1).
HIER([EMPLOYEE,SKILL],2).

に対して

EMPLOYEE(NAME,SALARY,[SKILL]).

のような概念スキーマを作成する。ただし、リスト表現[]は不定繰り返し項目を示す。

次に、主体同志による階層について示す。

ｃ）１：Ｎ（Ｎ＝１を含む）の場合には、Ｎの方の主体が１の方の主体を単一項目として持つ。

ｄ）Ｎ：Ｍの場合には、どちらか一方が他方を不定繰り返し項目としてもつ。

例えば、EMPLOYEE,DEPARTMENTとも主体のとき

HIER([EMPLOYEE,DEPARTMENT],3).

に対して

EMPLOYEE:DEPARTMENT=N:1の対応であるからEMPLOYEEのファイルにDEPARTMENTの項目を加え、上述のファイルは、

EMPLOYEE(NAME,DEPARTMENT,SALARY,[SKILL]).

となる。

最終的に得られる概念スキーマの例を以下に示す。

SELL(DEPARTMENT,ITEM).
SUPPLY(COMPANY,DEPARTMENT,ITEM).
DEPARTMENT(DEPARTMENT,FLOOR).
EMPLOYEE(NAME,DEPARTMENT,SALARY).
ITEM(ITEM,TYPE).

この例は、検索要求文(a1)～(a5)から得られたものである。この例には不定繰り返し項目は含まれてい

ない。また、主体を識別するために、属性にNAMEを持っていない主体にはそれ自身を項目に加えてある。

5．4　ＩＮＱ・ＦＤＬの設計

前の処理で得られた概念スキーマとＩＮＱ・ＦＤＬを用いた論理構造とは、１：１の対応がつくので、ＩＮＱ・ＦＤＬへの変換は一意的に容易に行える。

6．むすび

ＩＮＱを用いてデータベースを構築する利用者を支援するシステムの中の、データベースの論理設計を支援するシステムについて述べた。本システムを使用するにあたって利用者が用意すべきものは、開発しようとするデータベースへの検索要求文の集合だけであり、システムの簡単な質問に適宜答えればよく、負担は極めて小さい。

本システムではデータのセマンティクスに関する一切の事前情報は仮定していない。従って、意味解析を行っていないため、必ずしも完全な論理構造が得られるとは限らない。しかしながら、得られた結果は、対象領域をかなり整理して概念スキーマに表現してあるものであるから、利用者にはその論理構造表現を容易に理解でき、誤りをたやすく指摘できると思われる。従って、最終結果を利用者が容易に訂正できる画面エディター機能をシステムに持たせることにより、本システムの有用性が現実の場面において発揮されるものと思われる。

現在、インプリメンテーションを大阪大学大型計算機センターACOS-1000上でShapeUpを用いて行っている。完成度は約80％あるが、エディターの作成も行いながら、システムの評価をしていきたいと考えている。

謝辞

日頃熱心に御討論頂いている大阪大学産業科学研究所上原邦昭技官に感謝します。

＜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 base 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8,9日,pp.49-58.

[2]Isomoto,Y. et al.:"Knowledge Based DBMS for Non-professional Database Administrators", Proc. COMPCON83,pp.514-522(1983).

[3]小林他:"検索要求文に基づくデータベース論理構造の自動設計",情報処理学会第27回全国大会,5K-6,昭和58年10月.

[4]日本電気(株):INQ概説書(1981)

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

[6]横田他:"拡張PROLOG(Shape Up)の実現について", 記号処理研究会資料20-3,昭和57年10月.

[7]Pereira,F.C.N. et al:"Definite Clause Grammars for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks ",Artificial Inteligence 13,pp.~~221~~231-278(1980).

[8]Codd,E.F.:"Further Normalization of the Date Base Relational Model", Proc. Courant Computer Science Symposium 6, Data Base Systems, pp.33-64 (1971).