

データベースの論理設計を支援する 知的インタビューシステム

川口 敦生 溝口 理一郎 山口 高平 角所 収
大阪大学産業科学研究所

インタビューは、「専門領域が異なる専門家同志が、一方の専門知識の他方への移行を目的として行なう対話」と捉えることができる。この見方によると、特にインタビューの聞き手の振舞いに関して自然言語理解、学習等に関連する多くの興味深い問題が見出される。そこで我々は、インタビューの具体例としてデータベースの専門家（聞き手）とデータベースの発注者（受け手）が論理設計作業時に行なう対話を取り上げ、検討を加えてきた。そして、データベースの専門家の振舞いをシミュレートする知的インタビューシステムI²Sを開発してきた。I²Sは、ユーザにインタビューを実施し、得られた情報をもとに構築するデータベースの概念スキーマを設計する。本稿では、まずインタビューについて検討を加えた後、I²Sの構成、質問戦略およびその実現法について述べる。また、知識獲得の分野でインタビューに密接に関連している二つのシステム(MORE, ROGET)を紹介し、I²Sとの比較、検討を試みる。

An Intelligent Interview System for Database Logical Design

Atsuo KAWAGUCHI Riichiro MIZOGUCHI Takahira YAMAGUCHI Osamu KARUSHO
ISIR, Osaka University
Mihogaoka 8-1, Ibaraki, Osaka, 567, Japan

Interview is regarded as discourse between two persons who aim at transferring expertise from one to the other. From this viewpoint, interview has many interesting issues concerning natural language processing and learning mechanism. So we have developed an Intelligent Interview System (I²S) for Database Logical Design in order to investigate those issues. In this report, typical issues in interview and the basic mechanism of I²S are presented. Moreover, two other interview systems for knowledge acquisition in expert systems, called MORE and ROGET, are compared with I²S.

1. はじめに

インタビューは人間が行なう知的活動の一つである。身近な例として、テレビのレポーターが行なうインタビューが挙げられる。また近年盛んに行なわれているエキスパートシステム構築作業における知識獲得作業は、知識工学者が専門家に対して実施するインタビューが中心となっている。

インタビューは様々な局面で多様な形態をとって行なわれるが、本稿ではインタビューを図1に示すように「専門知識の移行を目的とした自然言語による対話」と定義する。

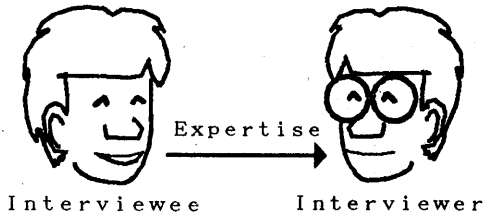


図1 インタビュー

この観点から特にインタビュアー (interviewer: 以下聞き手と呼ぶ) に注目すると、聞き手が持つべき機能として以下に示す4つのものがあると考えられる。

1. 自然言語による対話
2. 専門知識の理解
3. 専門知識の学習
4. インタビュー技術とその学習

専門知識の移行を目的とするため、対話中に聞き手にとって未知の概念が出現する。この未知概念について聞き手が理解、学習を行ないながら対話を進めねばならないのが、インタビューの特徴である。

本稿では、上記の聞き手に関する4要素について検討を加えることを目的として開発した知的インタビューシステム I²S (Intelligent Interview System) [Kavaguchi86] について報告する。まず次節で、知識獲得支援システム MORE と ROGET をインタビューの観点から概観する。次にデータベースの論理設計とインタビューの関連について述べる。4節ではシステムの概要を紹介する。そして5節で専門知識 (構築するデータベースのドメインに関する知識) の表現について述べた後、6節でインタビューを実施するための質問戦略およびその実現法について述べる。最後にインタビュー、現システムおよび今後の課題について検討する。

2. 知識獲得とインタビュー

エキスパートシステム構築作業における知識獲得作業は、知識工学者がドメインの専門家に対して実施するインタビューが中心である。したがって知識獲得支援システムは、多かれ少なかれインタビューシステムの性質を帯びている。本節では、インタビューシステムとしての性質が濃い二つの知識獲得支援システム MORE と ROGET を紹介する。

2.1 MORE [Kahn85]

MORE は、石油発掘に関連して掘穿泥水から地層を分析するシステムの考察から開発された「診断型問題に限定された知識獲得支援システム」である。

MORE は、3段階の洗練化操作を経てルールベースを構築していく。まず、専門家との対話を通して知識ベース第1版を構成する。この知識ベースは5つの実体からなるドメインモデルとして構築される。ドメインモデルの例を図2に示す。次に8つの戦略に基づいて専門家にインタビューを行ない、知識ベース第1版を洗練化し、第2版を構成する。最後に知識ベース第2版をルールベースに変換し、専門家に確信度を尋ねて、知識ベースの構成を終える。

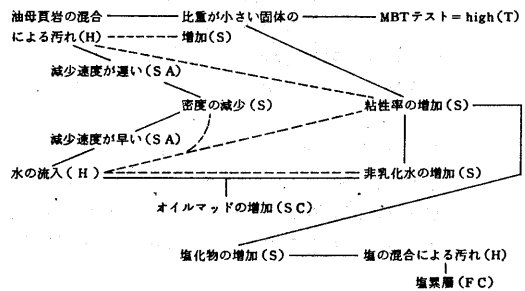


図2 ドメインモデルの例

ドメインモデルを洗練化する8つの戦略は、ネットワーク表現されたドメインモデルに対してノードおよびリンクの付加あるいは修正するためのメタレベル知識となっている。例えば仮設の区別 (Symptom distinction) と呼ばれる戦略は、全く同じ徴候 (S) を持つ仮説 (H) のペアに対して、それらを区別するための徴候を専門家に尋ねるものである。

MORE の特色は、ドメインモデルという中間的な知識ベースを介して、知識獲得 (インタビュー) の戦略を検討した点である。そしてこれらの戦略が診断型問題に対して広く適用可能であるというところに大きな価値がある。MORE の適用事例としては、ディスク故障診断、コンピュータネットワーク故障診断、回路ボード故障診断等がある。

2.1 ROGET [Bennet85]

ROGET は、診断型エキスパートシステムを試作する際に知識獲得支援を行なうシステムである。支援するのは、エキスパートシステム構築の初期に必要なドメインに現れる諸概念の整理であり、Conceptual Structure と呼ばれる表現を用いる。

ROGET は、次の4段階を経てドメインに現れる諸概念を整理する。まず、第1段階として目的とするエキスパートシステムのタスクの種類をユーザに聞く。そしてその種類にしたがって Conceptual Structure の雛形を用意する。次に第2段階では、この雛形の構造に基づいてユーザにインタビューを行ない、Conceptual Structure を構成する。

第3段階では、システムの資源、使用するエキスパートシステム構築支援ツールの特性等を考慮して、構築された Conceptual Structure を整形する。最後に完成した Conceptual Structure を、使用するエキスパートシステム構築支援ツールの表現形式で表現されたドメインの諸概念に変換し出力する。

図3に雛形の例を示す。これは、recommend-actions というタスクの雛形である。これに基づいてユーザにインタビューを行なって得られた Conceptual Structure の例が図4である。ここではMYCINの構成が例となっている。

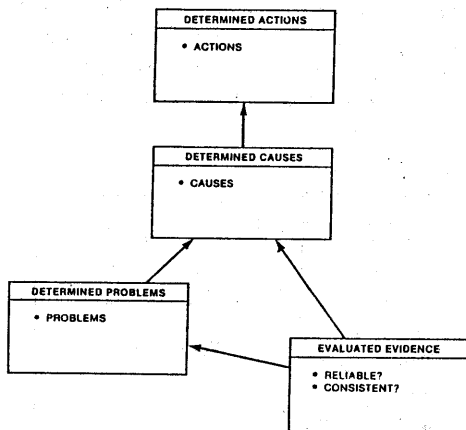


図3 Conceptual Structure の雛形

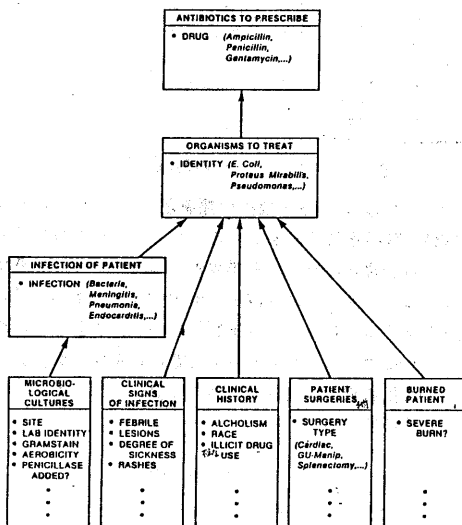


図4 Conceptual Structure の例 (MYCIN)

図3からわかるように、ROGETでは診断型問題に共通に現れる諸概念を分類、整理し、種々のタスクにおけるそれらの間の関連をあらかじめ知識として用意してある。

そして実際にエキスパートシステムの構築支援をする際に、これのスロットを埋める実体をユーザに質問していくことによってインタビューをおこなっている。すなわち、Conceptual Structure の雛形はあらかじめ質問すべき事柄を整理したものと考えられる。

3. データベース論理設計におけるインタビュー

本研究ではインタビューの例としては、データベースの論理設計作業におけるデータベースの専門家と発注者の対話を取り上げた。データベースの論理設計では構築するデータベースの論理的構造の決定を目標としているが、その作業の中心は構築するデータベースのドメインの的確なモデル化である。一般にドメインに関する詳細な知識は、発注者が持っている。したがって専門家は、発注者に多数の質問を発しながらモデル化を行なう。定義からこのような対話はインタビューである。すなわち専門家が聞き手であり、発注者にインタビューを実施しながら設計作業を進める。

本研究ではI²Sは、聞き手であるデータベースの専門家の振舞いをシミュレートするシステムの構築を試みている。このようなシステムの構築を通して1.に述べた聞き手の4機能について検討を加えていきたいと考えている。

またドメインの的確なモデル化という観点から、論理設計を意図したインタビューでは「良い刺激」の概念が重要である。ここで良い刺激とは「インタビューの受け手にとって曖昧あるいは忘れがちな事柄に気付かせる質問あるいは示唆」と定義する。聞き手が良い刺激を生成できれば、受け手も容易に必要な知識を提供できる。したがって良い刺激は、的確なモデル化を容易にし良い論理設計へとつながる。良い刺激の生成も本研究の重要な課題である。

なお、インタビューの一例として論理設計を取り上げたが、論理設計支援システムそのものもデータベース構築、運用をマニュアルレス化する際に欠くことのできないものである。これについては、7.1節で述べる。

4. システムの概要

I²Sは、以下に示す順序でインタビューを実施する。

- 1) 構築するデータベースの名前を聞く。
- 2) そのドメインの分類を聞く。
- 3) そのデータベースで処理したい検索要求文あるいはドメインにおける事実(以下要求文と略する)を聞く。
- 4) 要求文に関して質疑応答を行なう。
- 5) 論理設計を行ない概念スキーマを出力する。(関係型DBMSを使用することを前提としている。)

システムとの対話例を図5に示す。この例では、製造関係の会社のデータベースを論理設計するためのインタビューが行なわれている。

I²Sの構成を図6に示す。システムは大きく分けて4つのモジュールから構成されている。ユーザインタフェース、知識ベース、コントローラ及びメモリーである。これらのモジュールの内、メモリーが本システムを特徴付けている。

Please enter the name of your database.
 > maker_db
 What is the domain of the maker_db ? Please choose one from the followings :
 1 literature 2 company 3 school ...
 > 2
 Now, let's start designing conceptual schema of maker_db. Please enter an information requirement.
 > I want to know the names of parts which are constructed from other parts.
 Do you want to know the names of part-1's which are constructed from part-2's ?
 > Yes.
 Please enter next information requirement.
 > I want to know the names and quantities of parts from which another part is constructed and which are supplied from other companies.

Please enter next information requirement.
 > No more.
 Wait a moment. I think you should store the information about 'stock'.
 Do you agree ?
 > Yes.
 Please enter information requirement about 'stock'.
 > Some parts are stocked.
 Do you want to store the information about the fact "B units of part-1's are stocked" ?
 > Yes.

図5 システムとの対話例

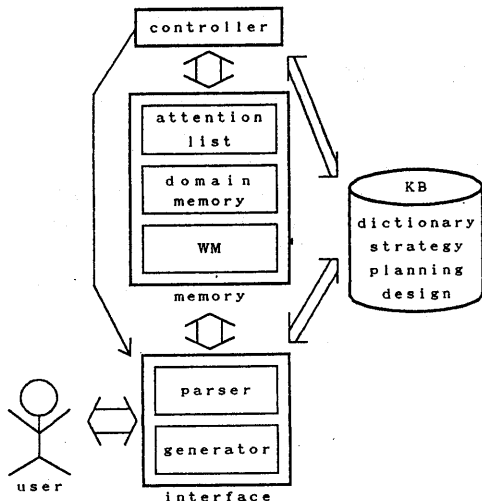


図6 システムの構成

メモリーは、domain memory, attention list および working memory からなる。domain memory は、ユーザから聞きだしたドメインに関する知識の格納に用いる。attention list は、対話中にシステムが気付いた注意事項 (attention) の一時記憶に、またWMは attention を処理する作業時に使われる。domain memory については次節で、attention list, WMについては6節で述べる。

3)4)の処理手順を図7に示す。入力処理でドメインに関する知識の内部表現を構成し domain memory に格納する。同時に attention が生成され(複数個)、attention list に格納される。個々のattention は、I²Sが持つ質問戦略の一つ一つに対応する。要求文の入力が終了すると、attention list 内の attention を順に処理して質疑応答を行なう。以上を繰返し行ない、最後に論理設計を行なう。

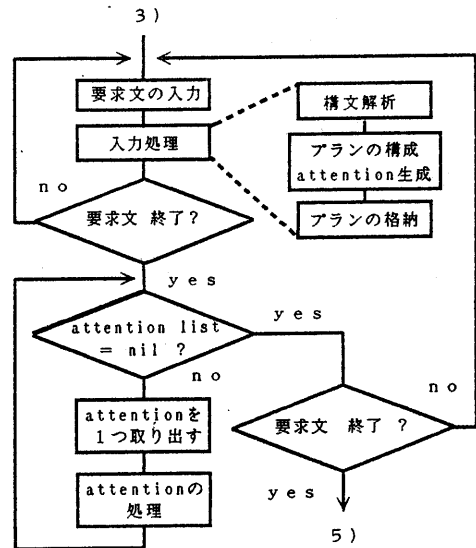


図7 インタビューの処理手順

表1にI²Sにおける知識の分類を示す。インタビューが専門知識の移行を目的とするため、システムがあらかじめ持っている知識とユーザから獲得する知識の切分けは重要である。本システムでは名詞を未知概念とし、それ以外の知識をシステムが持つことは制限しないということを前提とした。特にプランの概念を用いた動詞の意味知識およびいくつかの抽象概念に関する知識を用意し、これをインタビューの手掛りとしている。

表1 I²Sにおける知識

あらかじめ持っている知識	インタビューで獲得する知識
<ul style="list-style-type: none"> ・名詞以外の単語辞書 ・約30個の抽象概念に関する知識 ・動詞の意味辞書 ・構文知識 ・質問戦略 ・プランニングに関する知識 ・論理設計用の知識 	<ul style="list-style-type: none"> ・ドメインにおける実体や関連をプランの概念を用いて表現したモデル ・各実体(未知名詞)の概念区分 ・各実体(名詞)が持つ属性

5. プラン構造

I²Sでは、インタビューの対象となっているドメインにおける活動に関する知識は、プラン構造として表現される。

プラン構造では、個々の動詞、名詞が1つのフレームで表わされる。そして動詞の意味表現にプランの概念を導入している。名詞フレームにはその名詞の属性および上位概念が記述されている。フレーム間はポインタで結ばれ、全体でドメインにおける活動および事実を表現している。

動詞フレームは3つのスロット goal, action および precondition を持っている。各スロットはそれぞれ、その動詞が示す行為が達成する状態、その動詞が示す行為、その動詞が示す行為を実行するために達成されていなければならない状態を示している。各スロットを埋める要素として現在10個の意味素を定義して用いている。(表2) これらは、分類語彙表[国研64]から決定した。

表2 意味素

goal and precondition	
exist	:someone or something exists.
not_exist	:someone or something does not exist.
connect	:relation such as 'connect' exists.
action	
make_exist	:make someone or something exist.
make_not_exist	:make someone or something not exist.
make_connect	:make connection such as 'make a contract'.
make_not_connect	:make connection not exists.
change_how_many	:change quantity.
change_where	:equivalent to 'move', etc.
change_state	:change state of something.

ある動詞フレームAの goal と他の動詞フレームBの precondition が

「Aのgoalは、Bのpreconditionの一部をなしている」の関係にあるとき、AとBは接続可能であるという。さらに実世界で「ある行為Aを行為Bより前に行なわなければならない」ことが事実のとき、AはBに先立つあるいはBはAに従うという。一般には、二つ以上の動詞を順に接続することが可能である。そこでプラン構造では、一つ以上の(接続された)動詞フレームの集まりをプランと呼び、プラン間で「先立つ」「従う」といった接続関係を考える。そしてあるプランが与えられたとき、それに先立つプランあるいはそれに従う動詞を探索することをプランニングと呼ぶ。

動詞の表現にプランの概念を導入し、プランニングを行なうことによって、ドメインでの活動を推測することが可能になった。インタビューにおけるプランニングの使用については、次節に述べる。

一方、名詞フレームは super_class_is スロットと属性スロットからなる。super_class_is スロットには、知識ベース中の他の名詞フレームへのポインタが埋められる。すなわち、ある名詞はシステムにとって既知の名詞のサブクラスとして表現される。属性スロットは、その名詞が示すものが持つ固有の性質を示す。

ユーザの入力文中に出現した名詞がシステムにとって未知の名詞であった場合、その名詞フレームから知識ベース中の特別な名詞フレームである抽象概念フレームのどれかにポインタが張られる。(以下未知名詞のマッピングと呼ぶ) マッピングされた先の抽象概念に関する知識を用いてシステムはインタビューを続けることができる。[川口85]

プラン構造の例を図8に示す。この例は、図5の対話例に対応している。例えば、「ある部品は他のある部品から組立てられる」という活動は、矢印Aで指されたフレームで表されている。「部品」に関しては矢印Bが指すフレームがその属性を表現している。またこのフレームは、ポインタ(矢印C)によって抽象概念「物理的な物」にマッピングされている。

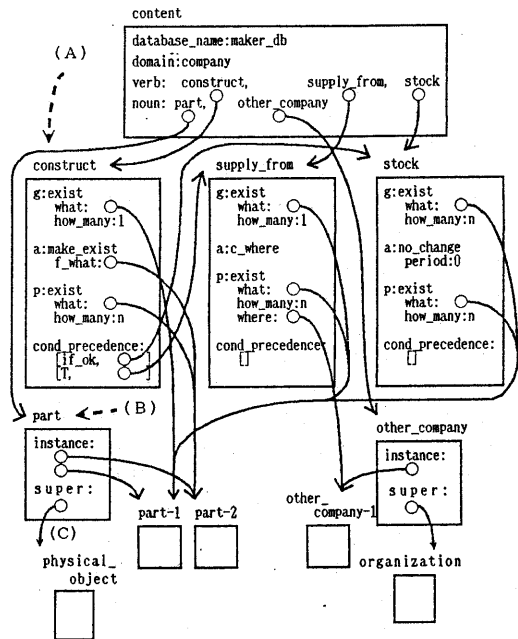


図8 プラン構造の例

6. 質問戦略とその実現

I²Sは、様々な質問戦略に基づきユーザと質疑応答を行なう。各々の質問戦略は、domain memory 内のプラン構造が持つ情報を豊かにする方向へシステムの動作を誘導するよう決定されている。

6.1 質問戦略

表3にI²Sが質疑応答時に従う質問戦略を示す。表中、attentionは対応するattentionを、また格納法はそのattentionが生成された際のattention listへの格納法を示している。この格納法は、attentionの処理順、すなわち質問の順序に関与している。詳細は、6.2で述べる。

表3 質問戦略

質問戦略	attention	格納法
1. 動詞フレーム単体について ・ PとGの詳細さを揃える。 ・ 時間に関するあいまいさを解消する。 ・ 再帰性を検討する。	check_detail ambiguity_period check_recursive	enqueue int enqueue
2. 一つのプランにおける動詞について ・ 動詞間の詳細さを揃える。 ・ 接続条件を検討する。	connect_detail connect_cond	enqueue enqueue
3. 複数のプランにおける動詞について ・ 現実世界で接続可能な未入力動詞をさがす。 ・ 同じ動詞間の詳細さを揃える。 ・ 現実世界で接続できる動詞の組をさがす。 ・ それらの組の動詞間の詳細さを揃える。 ・ それらの組の動詞間の接続条件を調べる。	search_unknown check_detail search_known connect_detail connect_cond	push enqueue push enqueue enqueue
4. 名詞について ・ 未知名詞ならば、抽象概念にマッピングし、属性を検討する。 ・ 既知名詞ならば、知識ベース中の名詞にマッピングし、属性を検討する。	map_unknown map_known	enqueue enqueue

質問戦略は以下に示すように大きく4つに分類される。

1. プランニング関連
2. 詳細度の統一
3. 名詞の取扱
4. その他

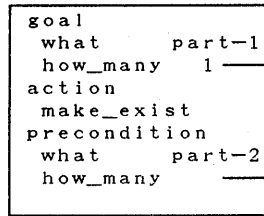
1,2は、プラン構造中の動詞フレームおよびそれらの接続に関して質問を行なう戦略である。3は名詞フレームおよびそれらのマッピングに関するものである。4は現在のところ、時間に関する質問を行なう戦略のみである。通常のデータベース管理システムでは一般に時間に関するデータを取り扱うのが困難なため、この戦略を用意してある。

戦略1は、プランニングを用いて、未入力（ユーザがまだ言及していない）動詞（活動）の必要性の示唆あるいは既知の動詞の接続を図るものである。一般に、ユーザの要求文がデータベース化する必要がある情報すべてについて言及していることは希である。この戦略によってユーザに刺激を与え、ドメインに関する知識をさらに引き出すことができる。

戦略2は、プラン中のgoalとpreconditionが持つ情報量を揃えようという戦略である。I²Sが行なう論理設計では、一つの動詞フレームが一つの関係（テーブルあるいは表）に対応する。そのため、この戦略に従うことによって完成したデータベース中の各テーブル間での結合演算が行ないやすくなる。

戦略1, 2の例を図9を用いて示す。図のconstructフレームは、ユーザが「1個の部品Aは、他の部品Bから組立てられる」という要求文を入力した際に作られる。プランニングを用いると、この例では動詞constructに先立つ動詞としてconstructやsupplyなどが見付かる。戦略1によって、このユーザが未だ入力していない（データベース化の必要性に気付いていない）動詞supplyに関する情報をデータベース化することをユーザに提案する。

Construct



② DETAILED LEVEL
"QUANTITY OF MATERIAL?"

① PLANNING

"CONSTRUCT, SUPPLY, etc.?"

図9 質問戦略1, 2の例

一方、goalおよびpreconditionを比較すると、図に示すようにhow_manyに関して情報量に差があることがわかる。戦略1によって、この例の場合材料である部品Bの数量をデータベース化するかユーザに問う。

戦略3は、システムにとって未知の名詞と既知の名詞を取り扱う場合に分けられる。未知名詞の場合[川口85]は、まずあらかじめシステム内に用意された抽象概念のどれかにマッピングする。次にその抽象概念に関する知識を用いて質疑応答を行なう。どの抽象概念にマッピングするかは、関連する動詞を手掛りとしてユーザに質問して決定する。既知名詞の場合は、名詞フレームを知識ベース中のそれに対応する名詞フレームにマッピングし同様に質疑応答を行なう。

6.2 質問戦略の実現

例として、ユーザが「1つの部品Xは他の部品Yから組立てられる」という事実を入力した場合を考える。(図10 a) まず、DCGに基づく構文解析機を用いて構文木の抽出を行なう。(b) 次に抽出された動詞に対応する動詞フレームを用いて、この事実を表現するプランを作成する。個々の動詞フレームは、構文木から自身のスロットを埋める情報を取り出すための知識を持っている。まず、自身の複製（インスタンス）を作成した後、この知識を用いて、そのスロットを順に埋めていく。(c) この時、各スロットの付加手続きが発火し、このフレームに関するattentionを生成する。(d) 次に生成されたプランをdomain memoryに格納する。格納はcontextフレームのverbスロットにこのフレームを加えることによって行なわれる。この時、verbスロットの付加手続きが発火し、プラン構造全体に関するattentionが生成される。(e)

attention list は、stack と queue の性質を合わせ持ったメモリーである。生成された attention は表3の質問戦略に付記した attention の格納法に従い、attention list に格納される。また、attention list は同じ動詞あるいは名詞に関する同じ種類の attention を、マージする機能を持っている。したがってそのような attention は、常に attention list 中に一つしか存在しない。この機能によって、以後の質疑応答においてむだな処理を避けることができる。

さて、ユーザが要求文の入力を終了し質疑応答段階に入ると、各 attention を順に attention list から取り出し処理を行なう。この時、attention の取り出しは attention list の先頭から順に行なわれる。したがって、生成時に push された attention が先に処理され、enqueue された attention が後で処理される。attention の格納法は、先に処理した方がよいと思われるものに push を割当ててある。各 attention の処理法は、プロダクションルールで記述されており、(f) 前向き推論を行なうプロダクションシステムで処理する。

このように質問戦略の実現は、attention の生成およびその処理という二つの段階を経て行なわれている。

```
One part X are constructed from other part Y.
(a) 入力文

[sent, [np, [a, [one]], [n, [part, x]]],
 [vp, [be_v, [ts]], [v_pp, [constructed]],
 [pp, [p, [from]], [np, [a, [other]], [n, [part, y]]]]]]]
(b) 構文木

part x --> goal:exist:what
1 --> goal:exist:how_many
part y --> precondition:exist:what
(c) slot filler --> slot

attention(check_is_recursive, instance(construct,0))
attention(check_detail, instance(construct,0))
(d) construct フレームから生成される attention

attention(search_unknown, instance(construct,0))
attention(search_known, instance(construct,0))
attention(map_unknown, part)
(e) context フレームから生成される attention

if operator ([instance(Op, _)])
then iisPlanning(Op, CandidateList), /* Do planning */
modify_wa(1, candidate(Op, CandidateList)).

if candidate(Op, [])
then make_wa(end).
(f) search_unknown を処理するルールの例 (一部)
```

図10 質問戦略の実現法

7. 検討

7.1 I²S の特徴

I²S の特徴を以下に示す。

1. ドメインの表現にプラン構造を用いる。
2. 入力文 (複数) の処理 → attention の生成 →

attention の処理という過程の繰返しで、インタビューを行なう。

3. 任意の名詞に対応できる。
4. データベースの論理設計を支援する。

プランの概念を、ドメインの表現に導入したことによって、単なるスロットフィリングでないインタビューが可能となった[Bennett85]。プランニングを行なうことにより、I²S は動詞フレームの組み合わせ、未入力動詞などについて、ユーザに示唆を与えることができる。

またシステムは、attention の生成、処理という流れをとることによって動的に質問を生成する。このため、あらかじめ用意してある質問を順にするという方法に比して的外れな質問が出にくい。また一つの attention が一つの質問戦略に対応するので、新しい質問戦略の実験が容易に出来る。

抽象概念へのマッピングという方法で、システムは未知名詞を取り扱う。インタビューでは、未知概念の取扱いは避けられない問題であるが、この方法によってある程度の対応が可能である。

データベースに関する AI 研究は、自然言語による検索システムの研究が中心であった [Kaplan83]。論理設計を支援するシステムの例は、極めて少ない [Mizoguchi85] [Haas83]。しかしながら今後、計算機システムの普及、DBMS の発達に伴って (素人が) 大規模なデータベースを構築する場合が増えてくるとと思われる (例えば学術データベース [H79])。その場合、データベース構築、運用全般にわたってこれをマニュアルレス化することが、大きな課題となり [沼田86]、その意味で従来人間が担当してきた論理設計を代行し得る本システムは意義があると考えられる。

次に、本システムの現在の課題は以下の通りである。

1. 意味素の妥当性の検討
2. 学習能力の付与
3. 認知モデルとしての検討

意味素および抽象概念の分類は、分類語彙表 [国研64] によっている。出来る限りの一般性を持たせたつもりだが、評価が十分とは言えず課題となっている。学習能力はまったくないといってよい。認知モデルとしても、汎用性、学習能力に欠けるなど問題が多い。

現在、特に学習能力実現および認知モデル検討を目的とした I²S 次期バージョンを計画中である。長期記憶 (あるいは動的記憶 [Schank82]) の考え方を導入し、これら二つの課題へアプローチすることを考えている。

7.2 知的インタビューの汎用フレームワーク

本研究では例としてデータベースの論理設計を取り上げたが、インタビューは人間活動の様々な状況で様々な事柄を目的として行なわれている。そこで、インタビューの汎用フレームワークと目的に依存する質問戦略の切分けが重要であると考えている。このような切分けは、インタビューの認知モデルへも通じるであろう。

図11は、インタビューの聞き手のモデル化を試みた例である。この例では、聞き手を大きく4つのモジュール（記憶システム、推論機関、入出力インタフェース）からなると考えている。また、インタビューの際行なわれる質問を次のように5つに分類している。

- 「確認」 推論した事柄を確かめるために行なう質問
- 「あいまい」 受け手の応答が曖昧な時に行なう質問
- 「意外」 自分が知っている事実と矛盾することを聞いたときに行なう質問
- 「形式」 始めからしなければならぬとわかっている質問
- 「推論不能」 推論できなかったために行なう質問

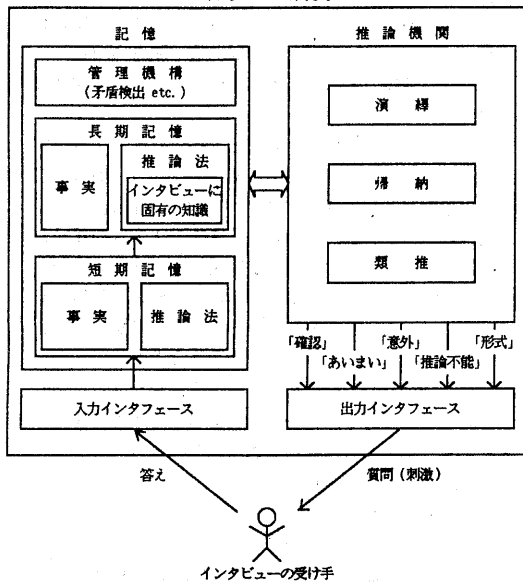


図11 聞き手のモデル

図のモデルによると、MORE, ROGET, I²Sは質問の性格から表4のように分類できる。

表4 質問の性格による分類

	MORE	ROGET	I ² S
確認	○	×	○
推論不能	○	×	×
意外	×	×	×
形式	○	○	○
あいまい	?	?	○

今後、図11のモデルをもとに、さらに検討を加え、知的インタビューシステムの汎用フレームワークを設計、インプリメントしていく予定である。フレームワークの評価は、

データベースの論理設計支援システムI²Sの実現の他、MOREやROGETあるいはこれらを更に一般化した漠然とした知識の整理支援システムの実現を通して行なっていく予定である。

8. 終りに

以上、インタビューおよび知的インタビューシステムI²Sについて述べた。また、今後の課題、予定についても述べた。

システムは、データゼネラル MV/8000 II 上の MV-Prolog でインプリメントされている。

参考文献:

[Bennett85] Bennett, J., "ROGET: A Knowledge-Based System for Acquiring the Conceptual Structure of a Diagnostic Expert System," *Journal of Automated Reasoning*, 1, 1985.

[Haas83] Haas, N. and Hendrix, G. G., "LEARNING BY BEING TOLD: ACQUIRING KNOWLEDGE FOR INFORMATION MANAGEMENT," Michalski, R. S., et al. (eds.), *Machine Learning*, Springer-Verlag, Berlin, 1983.

[Kahn85] Kahn, G., et al., "Strategies for Knowledge Acquisition," *IEEE PAMI*, Vol. PAMI-7, No.5, 1985.

[Kaplan83] Kaplan, S. J., "Cooperative Responses from a Portable Natural Language Database Query System," Brady, M. and Berwick, R. C. (eds.), *Computational Models of Discourse*, The MIT Press, 1983.

[Kawaguchi86] Kawaguchi, A., et al., "An Intelligent Interview System for Conceptual Design of Database," *Proc. ECAI'86*, Brighton, U.K., 1986.

[W79] 北川他, 文部省特定研究「情報システムの形成過程と学術情報の組織化」, データベース管理システム委員会53年度報告, 1979.

[Mizoguchi85] Mizoguchi, R., et al., "Interactive Synthesis of Conceptual Schema Based on Queries - Towards an expert system of relational database design -," *Journal of INFORMATION PROCESSING*, Information Processing Society of Japan, Vol. 8, No. 3, 1985.

[Schank82] Schank, R. C., *Dynamic memory*, Cambridge University Press, 1982.

[川口85] 川口他, 「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-」, *知識工学と人工知能*41-4, 1985.

[国研64] 国立国語研究所, 分類語彙表, 秀英出版, 1964.

[沼田] 沼田他, 「知的インタビューシステムI²Sに基づくデータベース構築支援システムの開発」, *情報処理学会第33回全国大会*, 4L-11, 1986.