

計算機援助型相談システム“CONSULTANT”の 構成とその設計思想†

磯本 征雄†† 石 柁 正 士†††
溝口 理一郎†††† 角 所 収††††

最近の計算機技術の向上に伴い、人工知能、データベース、知識ベースなどの知識集約をめざす研究が活発に行われている。これらに関連して筆者らも過去10年近く、計算機利用相談を計算機自体に行わせるためのコンサルテーション・システムの研究をすすめてきた。そしてCONSULTANTと名づけたシステムの開発を終了し、運用への足がかりをつかんだ。本相談システムでは、個々の相談内容特有の質問応答手順をモジュールにまとめ、相談に関する情報や応答手続きのデータベース化を試みた。本論文では、これら計算機利用相談システム構成の内訳とシステム設計における考え方について述べる。

1. はじめに

かねてより、汎用大型計算機の処理能力は増大しつづけ、高度な利用技術が次々とサービスされてきた。しかし一方ではそれらを有効に使いこなすために、高度で多量の知識が要求されることになり、計算機利用者の負担も増大しつづけている。超LSIに代表される最近のハードウェア技術の進歩に伴い、この傾向はますます強まるであろう。その結果、たとえ正規の情報処理教育を受けていても、計算機の利用に関しては、消化不良を起こしている者も少なくない。このような状況をふまえて、従来より各大学等にある共同利用計算センターや情報処理センターでは、毎年訪れる初心者への指導や新たに強化された機能の利用促進のために、“プログラム相談員制度”によって計算機利用の現場で生ずる諸問題解決の支援を行い、かなりの効果を収めている。

ところが多様な機能を相談員各員がすべて指導することは困難なので、これらを計算機の支援の下で行う試みが10年前頃から各センターで研究されてきた^{1)~6)}。しかしそれらは、「FORTRAN プログラムのデバッグ」のみとか「用語説明」のみに限定された断片的なものであった。このような状況のなかで筆者ら

は、最近のデータベース技術や知的システムの研究を参考にしながら^{7)~12)}、計算機利用相談での問合せ内容や応答形式などについて、特徴のあるいくつかの事例を同時に取り上げてこれらを鳥かんし、計算機による相談受付の機械化に関する諸問題を系統的に研究してきた^{13)~15)}。そして本相談システム開発のねらいは、大型計算機の豊富で優れた機能を活かして、作業代行、情報提供、問題解決などの計算機利用相談を計算機自体に行わせ、計算機システムを自己完結的体系にまとめることにある。

近年ソフトウェア工学において、プログラムの構造、とくに手続きとデータとの関係について盛んに論じられている¹²⁾。上述の目標達成をめざして本相談システムの構造についても、単純なデータ処理から高度な相談に至るまで、多様で効率的な各種モジュールの実現を図るために、システムの設計・構築に際して次のように機能を分離考察してきた：

- ① 相談事項に関する情報・知識のデータベース化
- ② 相談事項に対する適切な情報検索技法の適用
- ③ 相談内容に最適な質問応答を実現するためのマン・マシン・インタフェースの実現

さらに、データと手続きの役割分担などが積極的に検討され、最終的には従来のエキスパート・システム^{8),9)}や知識工学におけるプロダクション・システム¹⁰⁾と類似の構造となった。

本論文では、知識工学における考え方の応用を通して、筆者らが開発した計算機利用相談システムにおけるデータベースの役割および手続きとデータの関係について論じる。また、相談システムのモジュール化を基本とした、相談過程の定式化や情報のデータベース

† The Structure and Design Philosophy of A Computer-Aided Consultation System 'CONSULTANT' by YUKUO ISOMOTO (Osaka University Computation Center, Osaka University), TADASHI ISHIKETA (Faculty of Engineering, Osaka Electro-Communication University), RIIICHIRO MIZOGUCHI and OSAMU KAKUSHO (Institute of Scientific and Industrial Research, Osaka University).

†† 大阪大学大型計算機センター

††† 大阪電気通信大学工学部

†††† 大阪大学産業科学研究所

表 1 計算機利用相談システム (CONSULTANT) の構成モジュールと、システム駆動方式、相談受けの型および相談の内容

Table 1 A table of modules of the consultation system “CONSULTANT.” The table shows functions of the modules and their categories from the viewpoints of the types of operations and consulting methods.

モジュールの駆動方式	相談受けの型	モジュール名	相談の内容
手続き主導方式	作業代行型	FLOWCHART	フォートラン・プログラムの流れ図の作図
		MATHTEST	方程式数値解法プログラムの数値誤差解析
		DBGUIDE	データベース構築の際のジョブ・デック自動生成
情報・制御分離方式	情報提供型	\$SCHEDULE	計算機ジョブ処理状況の案内
		NEWS	計算センターからのニュースの広報
		TEACH	TSS コマンド一覧とその内容説明
		LIBPROG	プログラム・ライブラリの紹介
データベース主導方式	学支習型	GUIDE	計算機専門用語の解説・学習支援
		ADVISOR	フォートラン・プログラム・デバッグの助言
	助言・示唆型	DB-MANAGER	データベース構築・管理の知的支援
		?	本相談システム利用に関する相談受け

化の考え方について述べ、同時にこれらの統合体としてのシステムの現状について紹介する。

2. モジュール構成と個々のモジュールの特徴

大型計算機が使いこなせるためには、利用者としては、さまざまな情報や知識をもっていなければならない。しかもそのような情報や知識を得るための方法は、個々の内容によって異なっている。このような状況に対して有効に対処するために、データ処理の専門化のためのモジュール構成による効率向上と、情報・知識のデータベース化による多様性への柔軟な適応力の実現を図った。表 1 は、本論文で取り扱う利用相談の内訳とモジュール、相談受けの型から見た分類、各モジュールの駆動方式を示す。各モジュールは、端末からのモジュール名の入力によって呼び出される (図 1 参照)。モジュールは、相談内容を内包的に記述可能な程度に分割し、一般化可能な部分をプログラム化したものである。以下本章では、個々の構成モジュールについて、それらの特徴を述べる。

```

SYSTEM? CONSULTANT           ①
*****
* WELCOME TO CONSULTANT *
*****
MODULE? ADVISOR               ②
--- START ADVISOR ---
:

```

図 1 相談システム CONSULTANT の TSS 端末からの呼び出し方法

段階①: 相談システムの呼び出し。この段階は、エディタやコンパイラの呼び出しと同じレベルである。
段階②: 表 1 中の任意のモジュールの選択が可能である

Fig. 1 An access method to the consultation system “CONSULTANT” through a TSS terminal.

Step ①: access to CONSULTANT. This step is on the same level as a FORTRAN compiler or an editor of the host computer.

Step ②: One of the modules in Table 1 can be called at this step.

2.1 作業代行型

FLOWCHART¹⁶⁾, MATHTEST¹⁷⁾⁻¹⁹⁾, DBGUIDE²⁰⁾ などにおいては、いずれも個々の目的と処理は明確である。各モジュールは、エキスパートの行っている手続きを正確にプログラム化したものであり、相談といった意味合いは非常に少ない。ただしデータ処理の手続きは複雑であるため、モジュール開発には固有の困難さが伴う。

2.2 情報提供型

いわゆる情報検索である。ただしその方式には、メニュー選択方式 (分岐選択方式) のようなものからキーワード検索方式までさまざまなものがある。

\$SCHEDULE, NEWS, TEACH などでは、元の情報が階層的かつ表形式によく整理されているために、メニュー選択方式が最適であった²¹⁾。すなわち、大分類から小分類に向かって、逐次、表の項目を選択してゆくうちに、目的の情報に到達するものである。この方式は、PLATO²²⁾ をはじめ多くのシステムにおいて実用化されている。

一方、LIBPROG¹³⁾ においては、ライブラリ・プログラムの属性を示すと思われるキーワード (たとえば、単精度、連立一次方程式など) を手掛りにして、目的の副プログラムを探り出すものである。この方式は、通常の情報検索システムにおいて実用化されているものであり、ここではその技法をそのまま採用した。

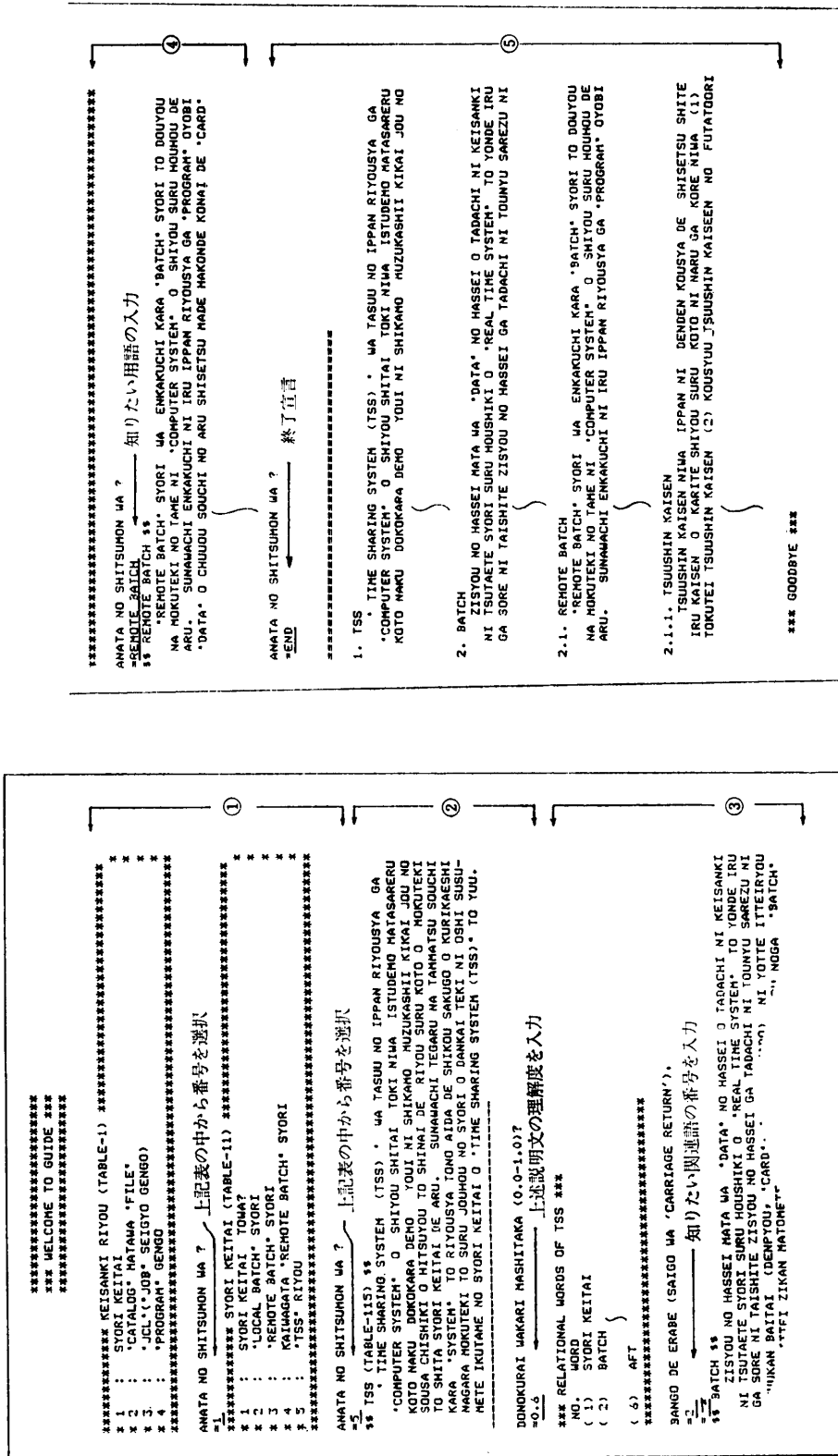


図 2 モジュール GUIDE の使用例 (下線部が利用者の入力)

①: テーブル (目次) から項目番号を選択。
②: 項目選択の最終段階での解説文の出力とこれに対する相談者の理解度を入力。
③: 未知の関連語の入力とその解説文の出力。

④: Selection of a number from the table of contents,
⑤: Explanatory sentences after a conclusive selection, and a consultant's response about his understanding.
⑥: Input of a consultant's unknown relative word and explanatory sentences about it.

④: キーワード入力とその解説文の出力。
⑤: 終了宣言 (END) とそれに対する処置としての解説文全体の章節立てによるまとめ。

Fig. 2 An example of use of GUIDE.

2.3 学習支援型

計算機利用者がその正しい基礎知識をもつことは、利用上の前提である。しかし時と場合に即して適切に対処してゆくには、かなり完成された知識をもつ必要がある。したがって十分な知識をもたない利用者は、自分の当座のジョブ処理に必要な知識を、数冊のマニュアルから拾い読みすることが多い。専門用語の解説システム GUIDE は、このようなマニュアルの拾い読みの学習支援を目的としたモジュールである。GUIDE への問合せの過程で、相談者に情報が与えられるだけでなく、誤った知識の訂正や、正しい知識の最適な再構成が可能である。

図2は、GUIDE の使用例を示す。図中①は、目次を繰り返しながら必要な情報を探索する操作に相当する。②は、目次から本文へと移ったことに相当する。③では、②を十分に理解するうえで必要な関連語が示され、相談者の知らない用語が表中にあれば、これを数字で指定することによってその用語の解説文が得られる。④における単語入力は、それまでの解説文中に現れた不明な用語の説明の要求である。

さて、②、③および④の解説文出力の直後には必ず、
DONOKURAI WAKARI MASHITAKA
(0.0-1.0)?

の質問が発せられ、相談者の理解の程度が問われる。完全に理解できたと思えば1.0を、まったくわからなければ0.0を入力し、理解の程度に応じて0.0と1.0の中間値を入力する。ENDの入力により、⑤に示すようにこれまでの全解説文を再編集し、個々の相談者に合った解説文を出力して終了となる。この編集の際に、上位語を先に下位語がそれにつづいて出力され、また同位にある語では上述の理解の程度の入力値を使って、理解された用語を先にして理解の程度の少ない用語を後まわしにして出力する。これらは、ソースを背景にもった情報検索の応用である。

GUIDE の設計において配慮された事柄は、学習中の相談者の問題意識自体が混とんとしていることである。知識の整理には、相談者の思考だけでなく調査とそのまとめといった形式的かつ手間のかかる手続きが必要である。したがって、GUIDE と相談者の情報交換をできるだけ単純にしてコマンド操作の煩雑さを排し、相談者の学習過程を乱さないことが必要である。実際のところ GUIDE では、次の3種類の入力促進のみにとどめた。

(1) あなたの質問は? (図2のステップ②、④参

照)

(2) どのくらいわかりましたか? (図2のステップ②末部参照)

(3) 関連語のうち、未知のものの番号を入力せよ (図2のステップ③参照)

(1)では、目次(表)からの章節番号の選択、単語入力、編集要求(EDIT)、終了宣言(END)など、各瞬間の相談者の課題を自由に入力できるようになっている。しかも入力がこれらのうちのいずれであるかは、GUIDE 自体が判断して処置をする。(2)は、最後の編集のための参考情報の収集である。(3)の段階の入力は、(1)の入力結果に対する関連処置である。このように GUIDE は、相談者が彼自身の問題意識の掘下げに専念できるように支援し、円滑な学習の進行を助けるように配慮されている。

2.4 助言・示唆型

プログラム作成の過程において、文法上の誤りはすべて取り除いたにもかかわらず、プログラムが意図どおりに稼動しないことはよくある。幸い、計算機言語をFORTRANに限定するならば、その言語仕様とのかかりによって、デバッグの多くは類型化される。

これらデバッグに関する経験的知識をデータベース化し、これらをプログラム作成者への助言に役立てることができる。ADVISOR²²⁾は、文法上の誤りだけでなく、それら以外のプログラムの誤動作に対するデバッグの助言も与えるように設計されている。

プログラム・デバッグに関する助言は、大きく3段階に分けられる^{21), 22)}。第1段階では、プログラム誤動作の症状を観察することで、デバッグ手続きについての大雑把な見当をつける。第2段階では、前述の見当をもとに症状に関する問診を行って、適切なデバッグ手順の具体的な可能性を引き出す。この段階において、症状からバグ(原因)に至る逆因果関係の有向グラフが描かれる(図3参照)。ただし、グラフ上の各分岐には推移確率が付けられており、マルコフ確率論を仮定することにより、推移の状況は容易に計算できる²³⁾。第3段階では、上記グラフをもとに計算された推移確率の大きさを目安にしながら、デバッグの助言を行う。細部における具体的取扱いを別にすれば、CASNET⁸⁾のモデルにおける階層構造によく似ている。

いま、推移事象間の確率推移行列を P_T 、推移事象から吸収事象への確率推移行列を P_E 、初期分布を \vec{p}_0 とすれば、最終的な吸収事象への到達確率を要素とす

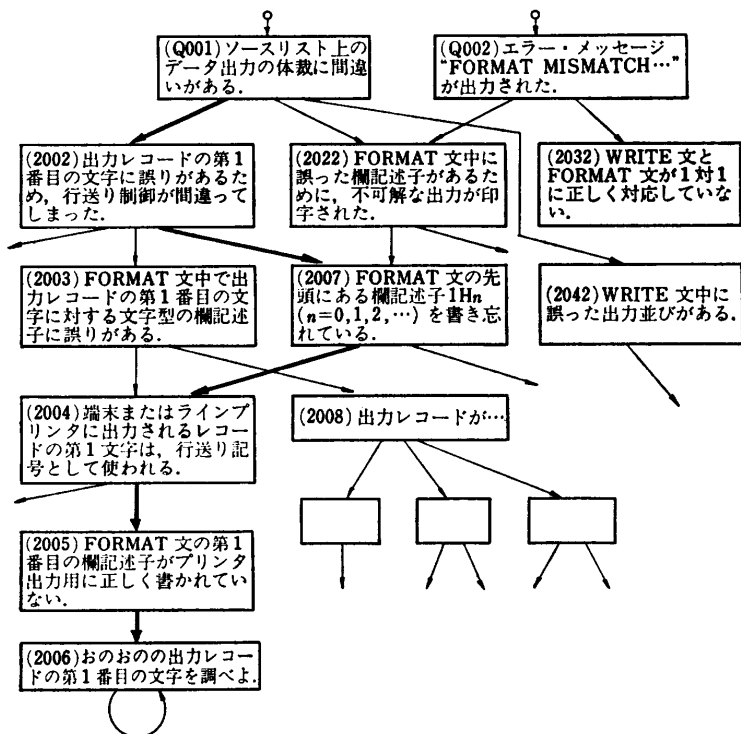


図3 ADVISOR で扱われる逆因果関係を表す有向グラフ
問診の後に最上位の事象 (Q001 または Q002) が一つ選ばれ、確率最大の順路 (たとえば太線の順路) が助言として出力される。

Fig. 3 An oriented graph that expresses the causalities from results to causes.
After diagnostic question-answering, ADVISOR determines a top vertex (for example, Q001 or Q002) of the oriented graph, and then generates an advisory message along the maximum probability path (for example, the path along bold arcs).

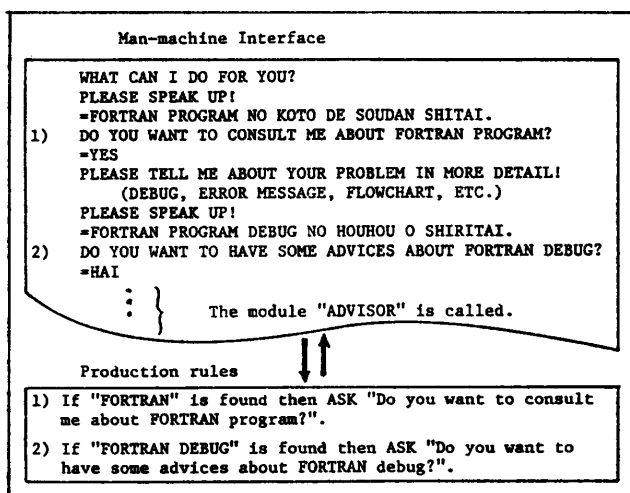


図4 モジュール“?”における会話例とその際に使われたプロダクション・ルール

Fig. 4 An example of man-machine interface between a consuler and the module “?”, and its production rules.

るベクトル \vec{q} は、次式で与えられる^{22), 23)}.

$$\vec{q} = \vec{p}_0 \cdot [I - P_T]^{-1} \cdot P_E$$

ただし、 I は P_T と同じ次元の単位行列である。 \vec{p}_0 における最大要素を i 、 \vec{q} における最大要素を e とする。このとき、 i から e への推移の確率 π_{ie} を次のように定義する。

$$\pi_{ie} = p_{ij_1} p_{j_1 j_2} p_{j_2 j_3} \dots p_{j_n e}$$

ここで、 i, j_1, \dots, j_n, e はおのおの順路上の事象であり p_{ij_1} 等は1段階推移の確率である。そこで最大の確率 π_{ie} を与える順路上の事象の連なり、 $i \rightarrow j_1 \rightarrow j_2 \rightarrow \dots \rightarrow j_n \rightarrow e$ をもって最も確率の高い助言とすることができる。

ADVISOR における助言の特徴は、密接に関連した事象間の関係を有向グラフと推移確率行列で表し、常に全体像を計算しながらすすめてゆくことである。すなわち、全体として確率的にしか関係を論ずることのできない FORTRAN プログラムのデバッグの議論などに適している。

2.5 問題解決型

人が物事の相談に向く場合、前述のようにすでに相談内容や視座・視点の定まったものは、その解決が比較的容易で

ある。しかし、しばしば自己の問題点すら理解されておらず、相談すべき事柄自体についても最初から相談しなせなければならぬことも多い。本相談システムのモジュール“?”では、自分の問題を解決してくれるモジュール名すら知らない相談者が本相談システムを使う際に、相談すべき内容や利用方法についての相談に応じることを試みた。

モジュール“?”を呼び出した後に、相談者が簡単なローマ字つづりの日本語か英文で相談事項を入力すれば、“?”は質疑応答をくり返ししながら、本相談システムのなかの必要なモジュールを示唆・提示し、その利用へと導く。これを“?”側から見れば、相談者がもつ問題とシステムが応じることのできる事項とのつき合せを行うことであり、その知的行動は、Weizenbaum の ELIZA⁹⁾ に類似した手続きのプロダクション・ルール¹⁰⁾によって制御される。図4は、モジュール“?”の

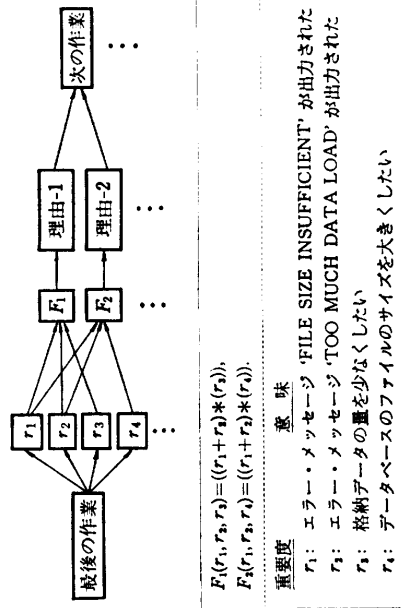


図 6 DBMANAGER における作業終了の状況から次の作業決定への理由付けの定式化

論理積 $r_1 * r_2 = \text{Min}(r_1, r_2)$
 論理和 $r_1 + r_2 = \text{Max}(r_1, r_2)$
 否定 $\neg r_1 = (1 - r_1)$

Fig. 6 Formalization of reasoning procedure of DBMANAGER to determine the next work from results of the last work.
 The reasoning procedure determines the next work by logical operation of fuzzy algebra:
 logical product; $r_1 \times r_2 = \text{Min}(r_1, r_2)$
 logical sum ; $r_1 + r_2 = \text{Max}(r_1, r_2)$
 negation ; $\neg r_1 = (1 - r_1)$

```

(MODE) E: EDIT R: RUN D: DIAGNOSE L: LIST
(JOB) (1) CREATE (2) INITIALIZATION (3) CONCEPTUAL
(4) INTERNAL (5) DATA COMPRESSION (6) DATA LOAD
(7) DELETE (8) RELEASE A LOCK (9) OTHERS
-----
SELECT ONE OF THE FOLLOWING WORKS; E5, R5, E6.
-----
THERE ARE ERROR MESSAGES ON YOUR LISTING SHEET FOR DATALOAD.
(1) FILE SIZE IS INSUFFICIENT.
(2) TOO MUCH DATA LOADING.
ANSWER THE FOLLOWING QUESTIONS.
Will you decrease the bulk of stored data?
-0-0 --- vNO vに相当する回答
Will you increase the file size?
-1-0 --- vYES vに相当する回答
** Let's update the file size of the database **
** PLEASE EDIT THE JOB DECK OF 'CREATE' **
-PS-/SIZE/ --- SIZEの文字列をもつ行にポイントをおわせる
FORCREATE/IDS DATABASE/TESTEDL,R,SIZE/0007/
-EVE/0007/;/0010/ --- 文字列0007を0010に変える
-RESAVE * --- 更新されたファイルを更新デバイス装置に格納
SAVED THE FILE CREATE
-----
SELECT ONE OF THE FOLLOWING WORKS; R1, E2.
-RL
* START THE WORK R1 *
SNOMB # 4047T
*JOB MONITOR*
4047T EXECUTING @ 10:27:00
4047T -001 WAIT-ALOC @ 10:27:11
4047T -001 EXECUTING @ 10:27:23
4047T OUTPUT WAITING ID=DF, @ 10:28:49
NORMAL TERMINATION
-----

```

図 5 DBMANAGER の稼働例
 システムからの質問に答えることにより、相談者は自分が次に何をなすべきかを定めることができる。
 Fig. 5 An example of use of DBMANAGER.
 Through question-answering, a consultant can determine his work to be done at the next step.

会話例とそこで用いられたプロダクション・ルールを示す。各プロダクション・ルールは、条件部と実行部から成る。“?”では、条件部にキーワードが書かれ、実行部には対話を行う関数名と会話の内容が書かれている。図4では、「If…」が条件部であり、「then ASK…」が実行部である。図4の入力文は、あらかじめ定められた条件部のなかの種々のキーワードとつき合わされ、その一致をもって対応するプロダクション・ルールが取り出される。

一方、DBMANAGERは^{24),25)}、データベース管理者がデータベース構築・管理、さらにデータベース更新作業を行う際に遭遇するさまざまな事柄や問題解決に対して知的支援を行う。データベース管理には、8組のジョブ・デックの編集と実行が必要である。とくに誤った操作を行った場合には、再度いくつかのジョブ・デックを実行して、データベースの再建をはからなければならない。この操作は、それ以前の作業履歴にも依存するために、多くの知識と経験が必要である。このような複雑な操作を支援するために、DBMANAGERは、ジョブ・デックの自動生成、ジョブ実行コマンドの自動発生、ジョブ履歴管理とそれらの制御を行う。そしてデータベース管理者は、MODEとJOBの組合せ(たとえばR1やE2など)のみを入力すればよい(図5参照)。

作業から作業への遷移には、ジョブ・デックの編集・処理状況、それら相互の論理的関係、管理者の意図などさまざまな事柄を配慮しなければならない。そしてDBMANAGERにおける履歴管理は、spiral modelによるscheduling map²⁶⁾上で行わせる工夫がなされている。次の作業を決めるにあたってDBMANAGERでは、“条件付け”、“理由付け”、“確証付け”の3段階の手続きがとられる。とくに“理由付け”の処理は論理演算の拡張であるfuzzy algebra¹¹⁾によって行われる(図6参照)。これらに関する熟練者の知識はデータベース化され、データベース管理者の作業上の課題解決のために使われる。

3. 本相談システムの設計思想

本章では、これまでに述べた個々のモジュールを鳥瞰するための筆者達の考え方を述べる。

3.1 相談システムの構造

相談システムの構造は、各モジュール内での“手続き(プログラム)”と“データ(データベース)”の関係によって理解できる。FLOWCHARTは、図示対

象のフォートラン・プログラムを入力データとして、流れ図作成手順をすべて手続き的に表現した典型例である。一方“?”は、手続き的内容の徹底したデータベース化を図り、手続き(プログラム)部分を最小限に留めた典型例である。このように各モジュールは、稼動時における処理効率向上をめざして、手続きとデータの適切な配分が工夫されている。

Wirthのプログラムの構造¹²⁾に関する模式化にならえば、上記のことは次のように表すことができる。

相談システム = 相談手続き + データベース (1)

相談手続き = 制御 + 手続き的知識 (2)

データベース = 手続き的知識 + 情報 (3)

制御; システム・プログラム (手続き的表現)

情報; データベース (記述的表現)

ここで手続き的知識というのは、情報の処理に関する知識であるが、これらを“手続き的表現”によってプログラム化することも、また“記述的表現”によってデータベース化することも可能な内容を指す。システム設計において問題になるのは、これら相談手続きとデータベースとの間の役割分担の境界である。すなわち、式(2)と(3)の手続き的知識をいかにうまく取り扱うかが、システム開発の重要課題である。表1における、システム構成の方式およびモジュール内での役割分担は、これらを典型的に示したものである。

作業代行型のモジュールでは、アルゴリズムの明確な規定が可能であり、データベースはいっさい用いずすべて処理はプログラムとして表現されているので、手続き主導方式と呼ぶ。情報提供型では、提供されるべき情報と情報提供の手続き(制御)が明確に分離されているので、情報・制御分離方式と呼ぶ。

学習支援型、助言・示唆型、問題解決型に共通していることは、提供されている情報の処理方法や、情報の提供に至るまでのシステムと相談者の対話の手続きが、個々の情報の内容に密接に関連していることである。この典型例は、DBMANAGERでの“最後の作業”の終了後にエラー・メッセージがあったことが知らされたうえで、相談者の意図を確かめて“次の作業”を決める」過程に見ることができる(図5参照)。この場合の対話の手続きは、エラー・メッセージの有無や相談者の意図に依存して異なってくる。

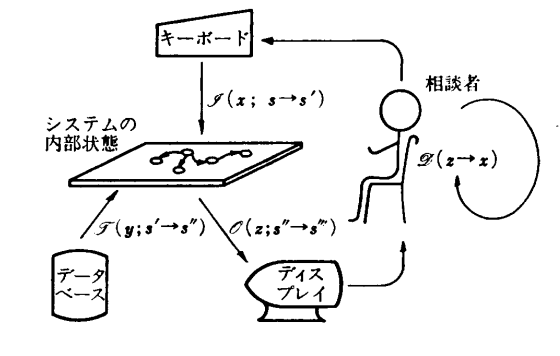
図6には、「対話での個々の質問応答がfuzzy algebraによって内部処理された後に、理由を付して相談者に次の作業の指示を与える」に至るようすを示した。“質問文”、“fuzzy algebra”、“理由付け”はすべ

てデータベースに格納されており、システム・プログラムは、データベースの検索と検索されたこれら手続き的知識の解釈と実行のみである（データベースの構造については5章で述べる）。したがってこれらを、データベース主導方式と呼ぶ。

相談システムの性能（知的レベル）は、それがもつ知識の量と質に大きく依存する。しかし計算機利用相談に必要な大量の情報を短期間に収集し、システム化することは容易でなく、したがって段階的に行わざるをえない。このことから、手続き主導方式のみでは、コードの修正というきわめて煩雑な作業を頻繁にくり返すことになる。そこでこれら3方式を使い分けることにより、システムのコーディングや保守を容易にするうえで大きな効果を上げることができる。さらに、データベース主導方式の導入により、新しい知識を徐々にデータベースに組み入れることで、複雑なシステムの段階的の水準向上が可能になる。

3.2 相談システム・モデルの一般化

システムのモジュール化の目的は、主としてデータ



記号	意味
$G(x; s \rightarrow s')$: 入力 x を得てシステム内部状態は $s \rightarrow s'$ へと遷移
$G(y; s' \rightarrow s'')$: データベースより情報 y を得て、システム内部状態は $s' \rightarrow s''$ へと遷移
$O(z; s'' \rightarrow s''')$: 出力 z を表示して、システム内部状態は $s'' \rightarrow s'''$ へと遷移
$D(z \rightarrow x)$: 情報 z を示された相談者は、内的変化の後にシステムへの新たな入力 x を作り出す

図7 相談システムの一般モデルを示す概念図
データベース内の知識を相談者に受け渡すが、本相談システムのねらいである。相談者とシステムはともに独立した内部状態をもち、会話を通して相互に影響し合う。

Fig. 7 A conceptual diagram that illustrates a general model of the consultation system “CONSULTANT.”

The system transfers information and knowledges in the database to a consulter. The communication between the system and a consulter changes their individual intrinsic states.

処理の専門化による効率向上である。しかし、将来にわたってモジュールの多様化によるシステムの機能向上をつづけてゆくには、全体的見通しの効く、総合的視点に立った、相談システム全体の一般的なモデル化が必要である。本相談システムでは、個々のモジュールの動きを、次の四つの基本機能を前提として、統一的に扱うことができる。

- ① 相談者からシステムへのデータ入力; G
- ② システムとデータベースのデータ交換; G
- ③ システムから相談者へのデータ出力; O
- ④ 相談者側の意識・認識の変化; D

図7は、データの流れから見た、これら4機能の相互の関係を示す。モジュールとしては、状態の変化 ($s \rightarrow s' \rightarrow s'' \rightarrow s'''$) とデータ (x, y, z) の流れを、単純な機能 (G, G, O) のくり返しの中で実現することで、システム全体としての構成の簡素化を図っている。

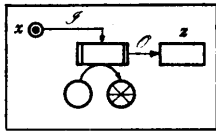
データ（情報や手続き的知識）は、これらの流れの中で加工処理され、変形・整理されながら相談が進められる。図8は、個々のモジュールを相談受けの型およびモジュール駆動方式の面から見た、 x, y, z の相互の関わりを示す。通常、相談者の意図の入力 x は比較的少量（1単語からたかだか数単語程度）であり、大半のデータはデータベースからの入力 y として受け入れられる。

3.3 知識ベース・システムとの関連

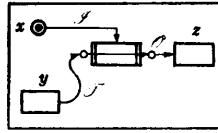
本相談システムは、いわゆるエキスパート・システムとか知識ベース・システムなどと多くの点で共通点をもつ。知識工学の分野においても、MYCIN, CASNET 等の医療診断システムに代表されるコンサルテーション・システムに関する研究が盛んに行われている。MYCIN¹⁰⁾ では、専門知識が situation-action の形のルールで表現されているところに特徴がある。MYCIN では、ルールの後向き推論プロセスの方式がとられている。すなわち、最初にゴールが与えられると、それを満足する action 部を有するルールが検索され、続いてそのルールの situation 部がサブゴールとして設定され、適宜質問がなされつつ以下この操作が繰り返される。この手法は通常の前向き推論とともにプロダクション・システムとして知られており、種々のシステムに応用されている。“?”や、DBMAN-AGER におけるジョブ・ステップ終了後の次のジョブ・ステップの選択機能の部分では、プロダクション・システムに類似した技法がとられている。

一方、CASNET⁸⁾ は、患者の診察結果、病理、生

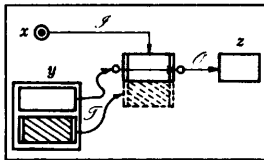
〔手続き主導方式〕
作業代行型 (FLOWCHART)
MATHTEST
DB-GUIDE



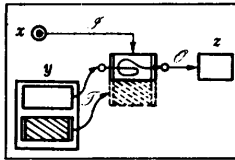
〔情報・制御分離方式〕
情報提供型 (SCHEDULE, NEWS)
TEACH, LIB-PROG



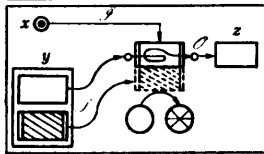
〔データベース主導方式〕
学習支援型 (GUIDE)



助言・示唆型 (ADVISOR)



問題解決型 (DB-MANAGER)



- ⊗ データの加工
- 相談者の要請事項
- モジュール(制御部)
- ▭ 情報の単純通過
- ◌ 情報の加工処理
- ▨ 手続き的知識
- ▩ モジュールに組み込まれた手続き的知識
- 情報

図 8 各モジュールごとに見た、1 サイクルにおける情報処理の特徴
Fig. 8 Characteristics of information processing of the modules through one cycle.

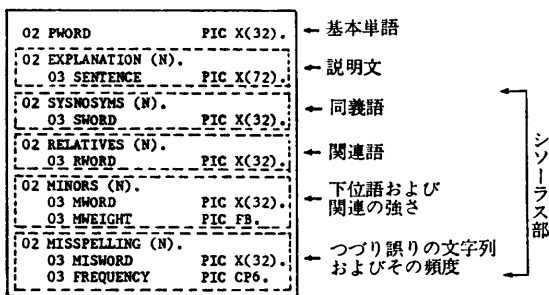


図 9 GUIDE のデータベースの論理構造
(N) の付された項目は、いわゆる "Repeating Group Item" と呼ばれているものである。

Fig. 9 Logical structure of the database of GUIDE.
An item with a tring '(N)' is so-called a "repeating group item."

理学的状態、疾患分類の三つのレベルのデータから成るセマンティック・ネットワークを知識表現のモデルとしている。本システムの ADVISOR では、まずデバッグの視点を与え、これに対する問診を行ってプロ

グラムの症状を診断し、その結果を用いて因果リンクからデバッグ手続きの助言を行う。情報処理の詳細な差異を無視すれば、ADVISOR と CASNET は共通の特徴をもっている。

これら知識ベース・システムに共通していることは、筆者らが作成してきた相談システムと同様に、蓄積される専門知識とそれらを扱う手続きとを明確に分離すると同時に、手続きをできるだけ汎用性のあるものとして、問題固有の知識はすべてデータベース化しようとしていることである。このように、専門知識あるいはシステムの行動を記述する基本単位をデータベースに格納しておき、手続きとしては状況に適した知識を検索し、それを解釈・実行するだけという検索→解釈→実行の繰返しによって問合せ処理を進めてゆく方式はかなり一般性があり、相談システム構築のための方法論として有望なものと考えてよいであろう。

4. 知識・情報のデータベース化

相談者に伝達される具体的情報や相談推進のための知識を格納するデータベースは、前章までに述べたモジュールの構造と表裏一体をなす。とくにモジュール側において吸収さ

れた内包的事柄に対して、データベース側は具体的かつ個別的な外延的事項を格納している。そして表1に示したように、データベースをもたない手続き主導方式からデータベースを主体としたデータベース主導方式に至る各種モジュールにおけるデータベースのはたす役割はさまざまである。さらにデータベース主導方式においては、データベースの論理構造によってモジュールの動きの主要部分が規定されてしまうといっても過言ではない。

さて、本相談システムのデータベースにおける最大の課題は、手続き的知識をできる限り一般的な形式でデータベース化することであった。したがって以下では、紙数の制約もあるので、データベース主導方式のみについて考察する。データベース主導方式では、原理的には相談に直接かかわるすべての知識がデータベース化されている。その際にとくに重要なことは、質問-応答の手順に関する知識のデータベース化に際して、できる限り一般的内容を包含しうるデータの論

理構造を定義することである。本章では、GUIDE, ADVISOR, DBMANAGER について、以下に具体的に述べる。

図9は、GUIDE (用語説明) のためのデータベースの論理構造を示す。用語解説文は、相談者の入力するキーワードに直接答えるためのものである。シソーラスには、下位語、関連語 (ここでは、反意語、類似語等も含めた)、同義語等が格納されている。この内容は直接に相談者に手渡されるものではないが、相談者の用語入力の多様性への対処 (同義語)、相談者の理解度の向上のための支援 (関連語)、会話終了時における個人用テキスト編集基準 (下位語) などとして、GUIDE の内部処理に用いられる。

図10は、ADVISOR (デバッグ助言) のためのデータベースの論理構造を示す。VIEWPOINT は、助言の開始時点での相談者のデバッグの視点設定の参考資料である。CROSS REFERENCE TABLE は、問診のための質問事項、問診結果から可能性の高いデバッグ技法を算出するための確率行列 C とから成っている。行列要素 C_{ij} 自体が知識の集約であり、これによりデバッグの助言の正しい方向が与えられる。ただし、この部分は質問者には直接見えない。事象間逆因果情報は、有向グラフの中の1事象の説明文とそこから次の事象への推移確率を与える。質問者へは、事

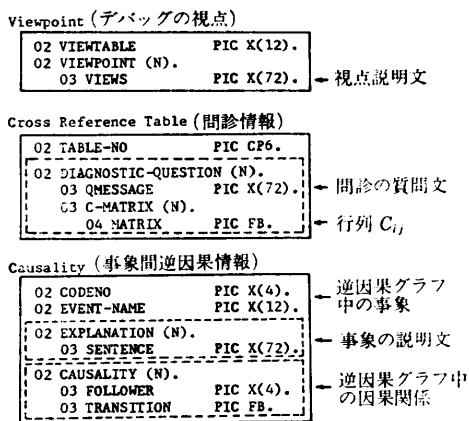


図10 ADVISOR のデータベースの論理構造
デバッグの視点、問診情報、図3に示された逆因果関係グラフの三つのサブファイルから成る。具体的内容は文献11)に示す。

Fig. 10 Logical structure of the database of ADVISOR.

The database is consisted of three subfiles; viewpoints of debugging, information about diagnostic question-answering, and causal network graphs from results to causes.

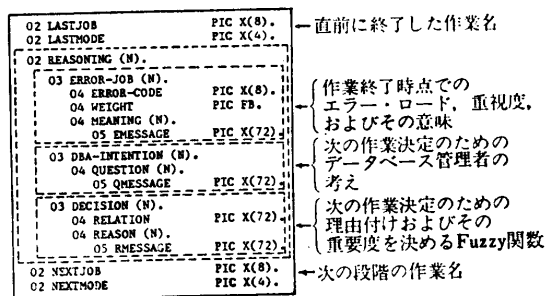


図11 DBMANAGER のデータベースの論理構造
主として図にかわる知識 (作業手順の推移の知識) が格納されている。

Fig. 11 Logical structure of the database of DBMANAGER.

象説明文のみが示され、他の情報は逆因果関係を示す有向グラフの生成とその再構成などに使われる。

図11は、DBMANAGER のためのデータベースの論理構造である。scheduling map 上での、作業 (LASTJOB と LASTMODE) から次の作業 (NEXTJOB と NEXTMODE) への進展は、主としてデータベースに格納された作業手順の理由付けに関する知識によって行われる。エラーの状況 (ERROR-JOB) とデータベース管理者の意向 (DBA-INTENTION) をもとに、fuzzy 関数 (RELATION) を使って、作業順序の理由付けの決定 (DECISION) がなされる。ここに格納される手続き的知識は、熟練したデータベース管理者によって確立されたデータベース構築・管理の専門知識である。

5. 実用性向上のための配慮

本相談システムは、当然、計算機に関する予備知識の少ない未熟練者によって補助的に使われるものである。したがって、本相談システムがその機能を十分発揮するためには、マン・マシン・インタフェースの機能に関して、システムからの回答の出力の形式、および相談者からシステムへの入力データの受入れ方法に対する配慮が必要である。

5.1 システムからの回答の出力形式への配慮

相談システム開発において、常識的ではあるが重要かつ基本的な原則は、相談者のもつ知識や問題意識、さらには視座・視点といった事柄に含まれるあいまいさを明瞭にしてゆく過程で、これに適切に対処することである。情報検索とは違って利用相談では、相談を推進する過程自体も相談の対象となる。相談者にとって、相談の結論はしばしば予想外のものであり、また

その当否は相談者自身の判断に委ねられている。さらに、複雑な思考過程で要求される質問-応答に対しては、“固定的かつ断定的な応答”を行うと、多くの場合、誤った結論に導くおそれがある。それゆえに、助言や示唆などの形式で相談者の思考を刺激しながら、質問-応答を繰り返して要求に合った回答に逐次近づける工夫が必要である。

このような事柄に関連し、計算機利用相談システムから相談者に示される回答の出力形式に関して、筆者らは次の三つの項目に整理し、配慮してきた。

- ① 相談者の不明瞭な相談意図の明確化：相談者は、しばしば質問事項があいまいなままで相談に臨むので、質問-応答の過程で相談対象を徐々に自覚するに至るよう誘導する必要がある。
- ② 質問から回答に至る手順の多様性：質問事項が明確であっても、具体的状況に対応した多種多様な回答の可能性があるために、試行錯誤的探索による回答への接近技法の工夫が必要である。
- ③ 回答の任意性：相談者への回答の妥当性は、彼の知的水準や、具体的状況、相談の目的など多くの要素に依存する。したがって、回答についてはこれらを配慮して何種類かを準備する必要がある。

以上の事柄に効果的に対処するために、本相談システムでは、相談内容を分析・分類し、データベースの内容を工夫し、相談者側の立場から見た適切なマン・マシン・インタフェースの実現を工夫してきた。

5.2 入力データ受入れの方法への配慮

未熟練者が端末を操作する際には、熟練者には想像しにくい多くの困難がある。したがって、多量のデータ入力を要求されるような形式での端末操作の方式は適切ではない。具体的には、「キーワードやパラメータ」入力などの単純な方法が望ましい。あるいは分岐選択、是非判定 (yes, no, unknown)、システム提示事項からの選択などの、相談者にとって受動的な応答のほうが実際的である。すなわち、相談システムの利用に習熟することを望むのは、現実的ではない。さらに、コマンド“?”をはじめとして、わからなくなった場合には相談者はいつでも記号?を入力し、システムより何をすべきかの助言が出されることが必要である。また相談者は、しばしばミス・タイプや思い違い等の多くの誤りを犯す。したがって、システム側ではわずかばかりの入力ミスに対しては、これを何らかの形で訂正や応答をするくらいの親切さも必要であ

る。たとえば GUIDE では、ミス・タイプと思われる単語には正しいと思われるつづりを示唆するようにしてある。また、他のモジュールについても同様な示唆機能や再試行機能を付けている。

6. ま と め

以上筆者達の開発した相談システムを概観し、他システムとの比較をも交じえて、それらに対する共通した背景について筆者らの見解を述べた。知識や知能をもとにした人間の知的活動の計算機によるシステム化の指針を明らかにし、今後このような複雑なシステム開発の方法論を明確にしてゆくうえからも、ここに述べた一般化のための議論が必要である。

ある特殊な事柄に限定した知的システムの議論は多い。しかし相談システムのような漠然としたシステムでは、そこで満たされるべき機能や内容は多様である。本論文では、モジュール化による相談システム開発の問題解決について述べた。一方、同時に、モジュール全体にわたる共通概念や相互関連を議論することにより、相談システムとしての統合された形態の定式化を試みた。これにより、さまざまなモジュール相互の役割と同時に、類似点や相違点を比較検討するうえで一つの視点を与えることができた。

マン・マシン・インタフェースにおけるあいまいさは、相談システムの常に問題とするところであり、その意味するところは複雑である。対象とする事柄に応じて、その内容や取扱いも異なる。本研究での筆者らの試みは、問題意識や心理状態の異なる質問応答過程の中でのあいまいさに対して、重要度、確率、fuzzy 代数等を取り入れ、異なった処理の仕方の比較検討を行った。さらにおのおのの機構の差異を越えて抽象化することにより、このようなシステムの開発における一般的技法が明らかになり、新しいモジュール開発に臨む場合の残された可能性の認識が容易になるであろう。今後計算機は、その性能・機能の強化によるサービス向上が続くであろう。その反面で従来どおりに、利用者自身が手引き書を読みながら自習によってそれらを使いこなす行き方には限界がある。したがってここで述べたような相談システムが将来ますます必要となるであろう。本論文は、このような将来の方向づけのための準備の一助となすためにまとめたものである。

参 考 文 献

- 1) 田中 一他: プログラム相談の機械化, 文部省科学研究費による特定研究「科学技術における学術情報処理例とその言語」研究報告 (1973).
- 2) 田中 一他: プログラム相談の機械化報告書, 文部省科学研究費による特定研究「広域大量情報の高次処理・科学技術における学術情報処理」(1976).
- 3) 藤井 博, 磯本征雄: プログラム相談支援のためのシステム開発, 全国共同利用大型計算機センター研究開発論文集, No. 2, pp. 151-159 (1980).
- 4) 赤石桂子他: HOSOS—プログラム相談のための情報コマンド・セット—, 情報処理学会第23回全国大会講演論文集, pp. 525-526 (1981).
- 5) 赤石桂子他: プログラム相談の機械化—相談システム(HOSOS)について—, 全国共同利用大型計算機センター研究開発論文集, No. 3, pp. 2-11 (1981).
- 6) 武富 敬他: 日本語によるセンター案内・相談支援システム, 全国共同利用大型計算機センター研究開発論文集, No. 4, pp. 109-114 (1982).
- 7) Hofstetter, F. T.: The Meaning of PLATO at the University of Delaware, *Computer Assisted Learning; Scope, Progress and Limits*, ed. by Lewis, Y. and Tagg, E. D., p. 123-134, North-Holland Pub. Co., Amsterdam, New York, Oxford (1979).
- 8) Sholom M. Weiss et al.: A Model-Based Method for Computer-Aided Medical Decision-Making, *Artif. Intell.* Vol. 11, No. 2, pp. 145-172 (1978).
- 9) Weizenbaum, J.: ELIZA—A Computer Program for the Study of Natural Language Communication between Man and Machine, *CACM*, Vol. 9, No. 1, pp. 36-45 (1966).
- 10) Davis, R. et al.: Production Rules as a Representation for a Knowledge-Based Consultation Program, *Artif. Intell.* Vol. 8, No. 1, pp. 15-45 (1977).
- 11) Zadeh, L. A.: Fuzzy Sets, *Inf. Control*, Vol. 8, No. 3, pp. 338-353 (1965).
- 12) Nicklaus Wirth: *Algorithms+Data Structures=Programs*, Prentice-Hall, New York (1976).
- 13) 石桁正士, 磯本征雄: 不特定多数の学習者を対象にしたコンピュータ援助型教育システム, 信学技報, ET 80-3, pp. 11-16 (1980).
- 14) 磯本征雄他: 相談・問い合わせシステム開発—データベース構築の面から—, 電子通信学会, 教育工学研究会, ET 81-1, pp. 25-30 (1981).
- 15) Isomoto, Y. et al.: A Computer Based Consulting System for Unskilled Programmers in University Education, *Proceedings of the Seventh International Conference on Improving University Teaching*, Japan, pp. 319-326 (15-18, July, 1981).
- 16) 岩川雅士, 磯本征雄: フォートラン・プログラムの流れ図・図示の一試み, 情報処理学会第17回全国大会講演論文集, pp. 605-607 (1976).
- 17) 磯本征雄他: 連立1次方程式に対する解法アルゴリズムの信頼性, 情報処理, Vol. 13, No. 9, pp. 593-598 (1972).
- 18) 磯本征雄他: 固有方程式数値解法プログラムのテスト方式, 情報処理, Vol. 16, No. 11, pp. 967-973 (1975).
- 19) Ohnaka, K., Isomoto, Y. and Makinouchi, S.: On a Test of Programs for Numerical Zeros of Polynomials, *Computing*, Vol. 25, No. 2, pp. 163-174 (1980).
- 20) Isomoto, Y. et al.: Guidance System for Structuring or Restructuring of a Database in Multiple Database Management Systems, *J. Inf. Process.*, Vol. 5, No. 3, pp. 182-187 (1982).
- 21) 磯本征雄, 石桁正士: CONSULTANT 計算機利用相談—利用説明書—, 大阪大学大型計算機センターニュース, Vol. 11, No. 2, pp. 120-132 (1981).
- 22) Isomoto, Y. et al.: A Computer-Aided Advising System for FORTRAN Program Debugging: ADVISOR, *Information Processing 80*, IFIP, North-Holland Pub. Co., pp. 985-990 (1980).
- 23) 磯本征雄他: 推移確率に基づく対談型事象検索システム, 情報処理, Vol. 19, No. 3, pp. 241-247 (1978).
- 24) 磯本征雄他: データベース・システム管理のための知的支援システム, 情報処理学会第25回全国大会講演論文集, pp. 695-696 (1982).
- 25) 磯本征雄他: データベース構築管理のための知的支援システム, 「アドバンスト・データベース・システム」シンポジウム予稿論文集, pp. 49-58 (1982. 12).

(昭和57年11月22日受付)

(昭和58年2月14日採録)