



## LISP教育用ICAIシステムの開発

その他（別言語等） のタイトル	The Construction of the ICAI System for Tutoring LISP
著者	伊藤 寿勝, 島本 肇, 黒島 利一, 杉岡 一郎
雑誌名	室蘭工業大学研究報告. 理工編
巻	39
ページ	17-26
発行年	1989-11-10
URL	<a href="http://hdl.handle.net/10258/768">http://hdl.handle.net/10258/768</a>

# LISP 教育用 ICAI システムの開発

伊藤 寿勝・島本 肇・黒島 利一・杉岡 一郎

## The Construction of the ICAI System for Tutoring LISP

TOSHIKATSU ITOH, HAJIME SHIMAMOTO, TOSHIKAZU KUROSHIMA and ICHIRO SUGIOKA

### Abstract

Traditional computer-assisted instruction (CAI) has advantages and also has some of disadvantages that should be improved. To study intelligent CAI (ICAI) is regarded as a trial to better CAI by using techniques in knowledge engineering. The system that was constructed in this research consists of three different modules—a special knowledge module, a student model module and a guide rule module—to solve some problems of conventional CAI. For future researchers, the opportunity for engaging the study concerned in some topics in the field of artificial intelligence (AI) is increasing, and it will be necessary for them to learn LISP, one of the programming languages for AI. In consideration of this tendency, the purpose of this research is to construct the ICAI system for tutoring LISP language.

### 1. はじめに

現在、人間の知的活動をコンピュータ上で実現しようとする人工知能 (Artificial Intelligence : 以下 AI) の研究が盛んに行なわれており、<sup>1)–4)</sup> その研究の 1 つとして、コンピュータに何らかの知的振舞いを行わせるソフトウェアの開発がある。AI の研究において最も重要なツールは、そうしたソフトウェアを作成・解釈・実行する AI 用プログラム言語である。

AI 用プログラム言語には記号処理機能が要求されるが、そのような機能を有している言語は論理型プログラム言語とよばれ、その代表的な言語は LISP である。

社会の情報構造が複雑化している今日、これからの技術者は何らかの形で AI の研究に携わる機会が多くなると考えられるため、本学のような工科系大学の学生に LISP を習得させることは非常に大きな意義を持っている。

一方、高性能で低価格のパーソナルコンピュータが普及し、工学・ビジネス・教育を始めとする多くの分野で使用されている。<sup>5),6)</sup> 特に、教育の分野では、コンピュータによって学習を支援する CAI (Computer Assisted Instruction) の普及がめざましい。

しかし、伝統的 CAI はいくつかのメリットを持つと同時に、多くの解決すべき問題をも抱えている。そのため、伝統的 CAI に欠けている要因を知識工学 (人工知能) 的な手法によって解

決していこうとする ICAI (知的 CAI=Intelligent CAI) が<sup>7),8)</sup>、急速に注目を浴びてきた。

したがって本研究では、人工知能用言語と言われる LISP 言語を教育するための ICAI システムを LISP を用いて構築した。

## 2. 学習項目

本システムでは、できるかぎり LISP 言語全体を網羅することを目的として、下記の12の項目をとり挙げた。<sup>9)~13)</sup>

- |              |             |              |
|--------------|-------------|--------------|
| A. LISP とは   | E. 論理演算     | I. 繰り返し処理    |
| B. 基本的な LISP | F. ユーザー定義関数 | J. 入出力操作     |
| C. リスト処理     | G. 条件分岐     | K. 局所変数と大域変数 |
| D. 数値演算      | H. 再帰的定義    | L. 総合問題      |

この12のカリキュラムで、LISP とはどのようなものかを系統だてて把握できると考えられる。また、最後の総合問題では、「バックトラック法」を用いた N クイーンの問題によって、これらの項目で学んだ学習者の最終的な理解の程度を判断している。

## 3. システムの構造

一般に ICAI システムにおいては、①学習者の理解度を正しく把握する機能、②適切な問題を生成し出題する機能、③学習者の誤答の原因を同定する機能、④学習者からの質問に答える機能、⑤同定した誤り原因から教育的見地に立って学習者を指導する機能等の機能が要求される。

また、ICAI システムは、従来の CAI の欠点を知識工学的な手法を導入することにより解決しようとするもので、それを実践するためには、①専門知識モジュール、②学生モデルモジュール、③指導規則モジュールの3つのモジュールを作成することが必要である。3つのモジュールを上記の各項目に置くことによって、LISP 言語の習得度の向上を目指している。

図一1に、3つのモジュールを図式化したものを示す。

### ① 専門知識モジュール

これは、学習者に習得させたい知識である。本システムでは、LISP 言語の文法及び語句、関数の知識、またその項目での問題そしてその問題に対する解答群からなる知識から構成されている。問題の出題形式は、いくつかの選択肢の中から数字で選ばせるものや、特定の語句を入力させるもの、LISP 関数を実際に定義(プログラム作成)させるもの等が用意されている。また解答群はできるだけ、学習者が考え得る解答をカバーできるように多岐にわたっている。

### ② 学生モデルモジュール

このモジュールでは、コンピュータ内部で学習者の理解状態を正しく認識するために、その学習者がどの項目まで学習したかを示す学生モデルと、その学習者が個々の問題に対してどの程度

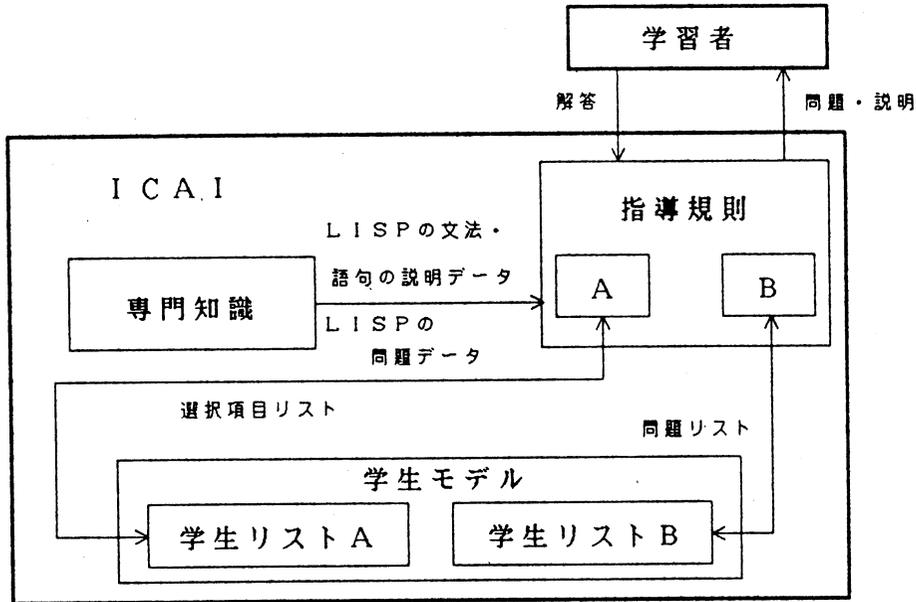


図-1 システムの構造

理解しているか、また問題のどこで間違えたかを認識する学生モデルの2つの学生モデルに分けて構成している。

上記の学生モデルは、それぞれ学生リスト A、学生リスト B として構築されている。

以下に学生リスト A の構成を示す。

(ID 番号 A 回数 B 回数 … L 回数)

ID 番号は、現在、学習を行っている学習者の ID 番号である。また、アルファベット A, B, …, L のそれぞれは、2. で述べた学習項目を示しており、各アルファベットの隣の回数は、その項目を学習した回数を示す。このリストを基礎としてどこの項目を学習すれば良いかの指示を出すことができる。

また、学生リスト B は A から E までの項目の学習時と F 項目以降の学習時とでは構成が異なる。これは、F 項目以降では学習者に LISP 関数を実際に定義させているため、より詳細に学習者の理解度を把握することができるようにするための配慮である。

以下に学生リスト B の構成を示す。

i) A~E の学生リスト B

(ID 番号 項目 NUM1 NUM2 NUM3 日付)

ID 番号は、現在、学習を行っている学習者の ID 番号である。項目には、A から E 項目に対応する英字が入り、NUM 1 はその項目で行った演習問題の問題番号で、NUM 2 は NUM 1 の問題の中の何番目の設問かを知る数字で、NUM 3 はその問題に対して間違えた回数を示す。また、日付はその問題を行った日付である。

ii) F~L の学生リスト B

(ID 番号 問題 MM 回数 M1 回数 M2 回数 … 日付)

リストの中の“問題”は、どの問題かを表し、MM の隣の回数は LISP 関数の定義中に学習者が全く理解していないと思われる入力を行った回数で、以下 M1, M2, … の隣の回数は、重要度順にどのような部分を何回間違えたかを把握するために表した数字である。また、日付はその問題を行った日付を表している。

### ③ 指導規則モジュール

指導規則モジュールでは上記の学生リストを使用する。学生リスト A で、どこの項目に学習者を誘導するのかを決定し、学生リスト B で、その問題に対する理解度を把握したり、また、学習者の間違えに対応したヒントの出し方を決定している。

指導規則 A は、学生リスト A の状態から、どこの項目に学習者を誘導すればよいかを決定する。これによって、例えば A 項目や B 項目を全く理解していない学習者が L 項目の総合問題を選択しても、A 項目や B 項目へ学習者を誘導することが出来る。ここで重要になってくるのは、何回この項目を学習したから次は、「この項目へ行きなさい。」という指示の出し方である。ここを強化することによって、項目別に弱点の補強ができる。しかし、この部分には、教育的な勧奨ともいべき経験則が適用されなければならないので、今後も研究の課題となる部分である。

指導規則 B は、学生リスト B の状態から、問題に対して学習者が間違えた場合のヒントの出し方を決定したり、その問題に対する理解度を把握する。また、この指導規則 B では、ある学習者がある問題についての正解率があるレベル以上であったならば、その学習者はその問題を理解したとみなして、次の学習の時には、その問題を出題しないようにしている。

ヒントの一例として、例えば関数 CUBIC を定義する問題で (DEFUN CUBIK…) と学習者が入力した場合、システムは「関数名はありますか?」というヒントを出して関数名が間違っている事を促し、学生リスト B に関数名の誤りがあったことを記録する。

また、実際に関数定義をさせているので、学習者が関数をキーボードから入力していて、途中でわからなくなった場合は、「WHY」と入力するとシステムは、次にどのようなことをすれば良いのかを学生モデルを分析して判断し、学習者に示唆する。

## 4. システムの解説

図-2 にシステムのファイル構成を示す。

LISP 教育用 ICAI システムの開発

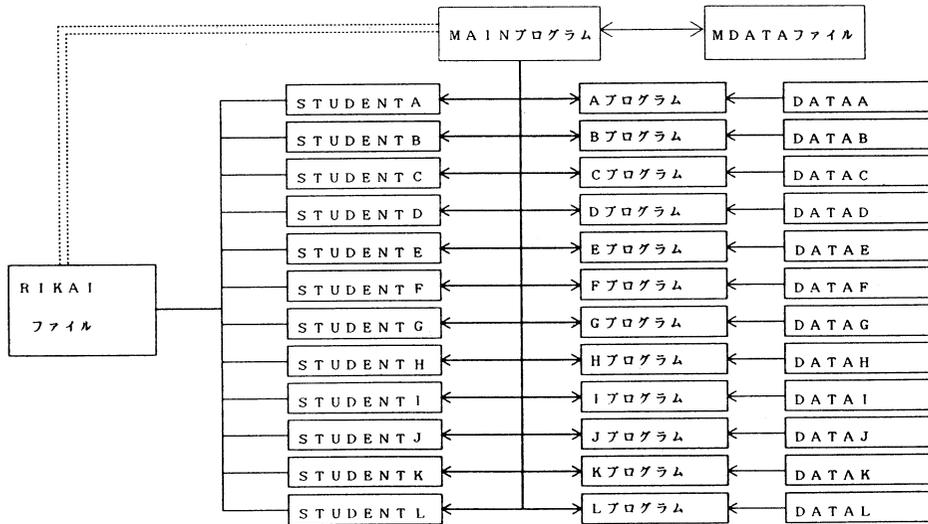


図-2 システムのファイル構成

MAIN プログラムは、システムの頭脳部であり、図-2のように全てのプログラムを統括している。処理内容は、①図-2のAプログラムからLプログラムのメモリーへの読み込み、②学習者に学習させたい項目の表示、③AプログラムからLプログラムの中で共通して使用されている関数の定義、④学生モデルの更新及び作成、⑤学生モデルを監視することによる個人指導、⑥各項目の問題に対する学習者の理解度の表示等を行っている。

DATAA~DATALには、その項目ごとの説明すべきデータ及び実行制御フラグが入っている。そのフラグを使用することによってAプログラム~Lプログラムに指示を与えることができる。例えば、DATA中に(改行)というフラグをプログラム側で認識すると、プログラム側では改行を行い、(演習)というフラグを認識すると、プログラム側では学習者に対して問題を出題するようになっている。

それぞれの項目のプログラム(Aプログラム~Lプログラム)は、DATAA~DATALから読み込んだデータをディスプレイに出力し、実行制御フラグによって、実行すべき関数に移行する。また、ディスプレイ上に表示すべき表などの関数を定義している。ここでは、それぞれの項目ごとに、その項目に適した数題の問題を用意してある。また、それぞれの問題に対する学習者の応答に応じて、数多くのヒントが用意されており、かつ問題の途中でわからなくなった場合、「WHY」と入力することにより、次に何をすべきかを学習者に提示する機能も備わっている。

MDATAファイルには、先に説明した学生モデルに2種類に分けて構成されている学生リストAと学生リストBの内の学生リストAが記憶されている。

STUDENTA～STUDENTLには、学生リストBが記憶されている。STUDENTAには、A項目の問題に対する学生リストBという具合に、ファイルにはその項目ごとのリストが蓄えられている。

RIKAIファイルでは、それぞれの学習者の項目ごとの問題に対する理解度を学生リストBを用いて作成し、パーセントで表記したものを蓄えている。またその問題を行った日付も一緒に保存している。

## 5. 利用形態

本システムは、電源を投入すると自動的にMAINファイルがローディングされて実行待機状態になり、学習者に「(ICAI)と入力してください。」と指示を与える。ここで学習者が(ICAI)と入力すると、システムが起動され初期画面が表示された後、学習者にID番号の入力要求をする。学習者がID番号を入力するとディスプレイ上に、学習項目を表示する。(図-3参照)

そして、学習者が自分の学習したい項目を選択すると、まずその学習者に対しての学生リストAを参照または新規作成する。その後、その学習者が過去に行った問題に対する理解度を表示する。(図-4参照)

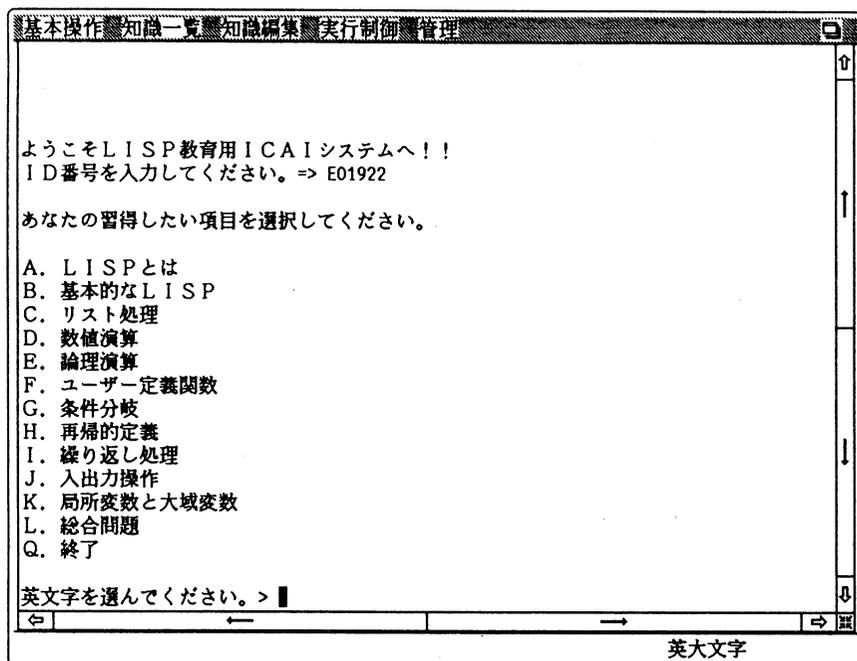


図-3 項目表示

基本操作 知識一覧 知識編集 実行制御 管理			
Sと入力してください。> s			
今までのあなたの理解度を表示します。(ID番号 E01922)			
A1の問題==>96 %	B1の問題==>100 %	C1の問題==>100 %	C2の問題==>95 %
D1の問題==>100 %	D2の問題==>100 %	D3の問題==>100 %	E1の問題==>85 %
E2の問題==>100 %	F1の問題==>91 %	F2の問題==>93 %	F3の問題==>96 %
G1の問題==>未実行	G2の問題==>97 %	G3の問題==>100 %	H1の問題==>95 %
H2の問題==>100 %	H3の問題==>0 %	I1の問題==>98 %	I2の問題==>100 %
I3の問題==>97 %	J1の問題==>未実行	J2の問題==>0 %	J3の問題==>未実行
K1の問題==>97 %	K2の問題==>未実行	K3の問題==>未実行	L1の問題==>0 %
L2の問題==>0 %	L3の問題==>0 %	L4の問題==>0 %	L5の問題==>0 %
L6の問題==>0 %	L7の問題==>0 %		
Sと入力するとスタートします。> █			
英大文字			

図-4 理解度表示

例えば、学習者がA項目を選択したとする。そうするとA項目の説明が始まる。説明は画面単位で進行され、一画面分の説明が表示されるとシステムは、説明を継続するか、あるいは終了するか、それとも説明を前に戻すのかを学習者に問うようにしている。これによって学習者は自分の意志で学習の進行を制御できる。また、進行中、問題が用意してある場所にくると、先の選択肢に加えて問題を行うか否かも問うようにしている。

問題に入った場合の一例として、ここではI項目(繰り返し処理)の問題を示す。この問題は、PROGとGOを用いて、階乗を求める関数の定義である。

問題では、何度かの解説及び説明を行った後、実際に学習者に関数を定義するように入力要求をする。学習者が入力した関数に誤りがある場合、システムはどこが間違っているのかを指摘する。(図-5参照)

また、学習者がどこをどの様に訂正すれば正しい関数を定義できるのかわからなくなった場合には、「WHY」と入力することによりシステムは有効な助言を与える。(図-6参照)

学習者が正しい関数を入力できた場合、システムは「きちんと定義されていますね。」というメッセージを表示し、この設問に対する学習者の理解度を表示し、次の問題に移行するかどうかを学習者に尋ねる。(図-7参照)

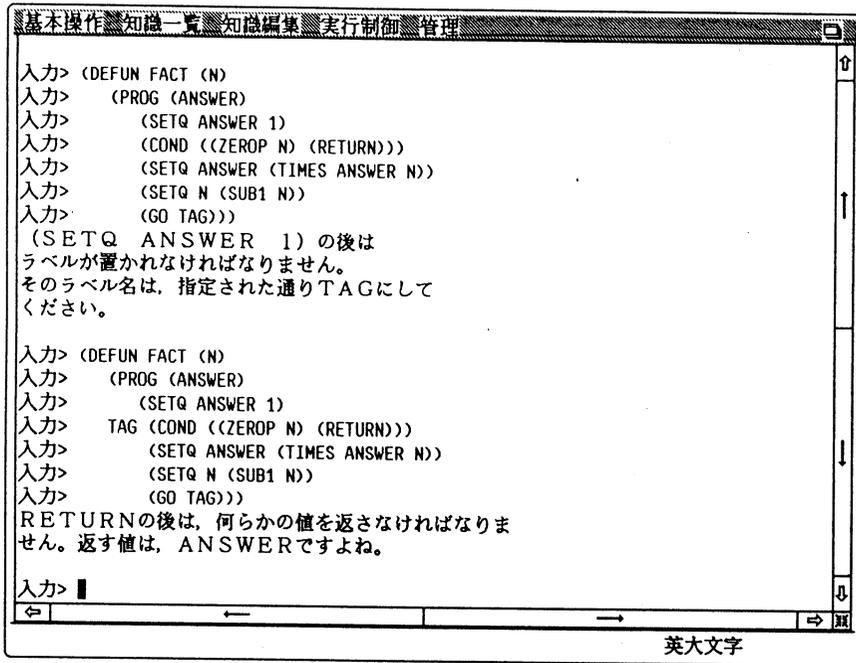


図-5 問題解答例

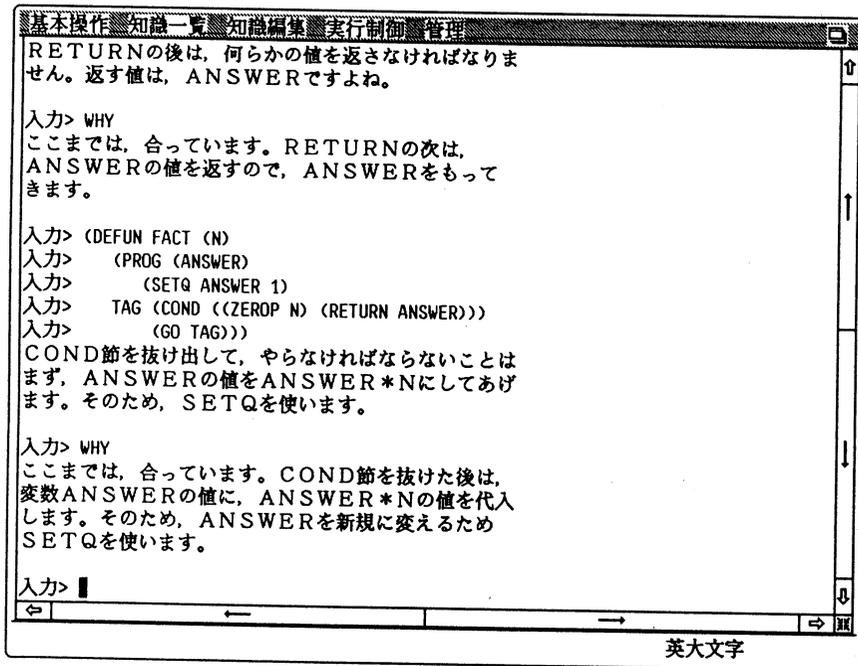


図-6 「WHY」と入力した場合

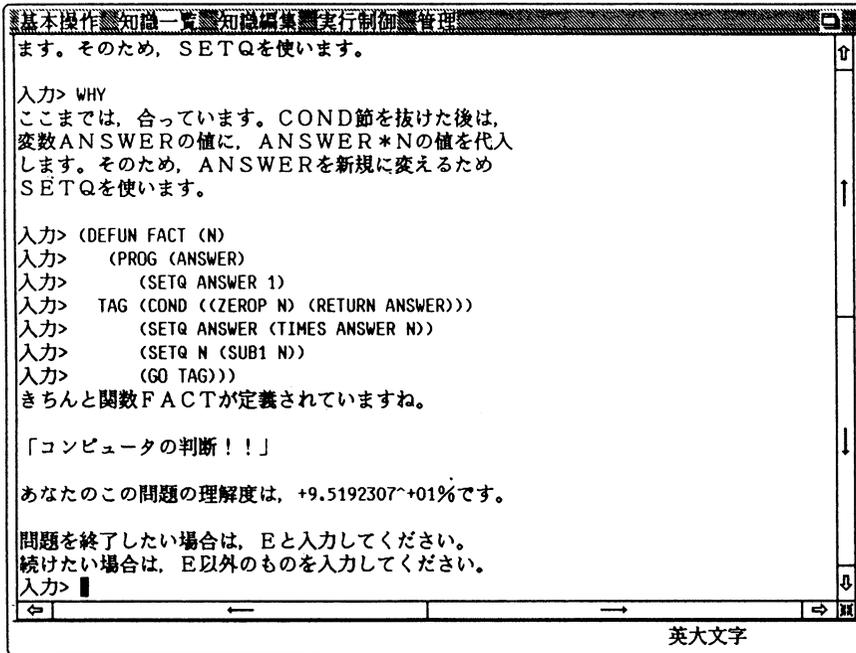


図-7 学習者の正解入力

このシステムをL項目（総合問題）の最後まで学習すると、LISP習得における総理解度を表示するようにしてあり、それに応じて自分の不十分な部分を何度も繰り返して学習できるようにしている。

## 6. おわりに

本研究では、LISP言語を用いてLISP教育用ICAIシステムの開発を行った。本システムでは、ICAIシステムに要求される機能の1つである学習者の誤答の原因を同定する機能を持たせた事により、ある程度の成果を上げることができた。また、学習者の簡単な質問には、部分的に答えられるようになっている。学習者の理解度を正しく把握する機能もICAIの大切な要素であるが、簡潔な方法であるパーセントで理解度を表示するという事で機能を満たしている。また、学習者の理解度に応じた学習の進行をある程度制御できるようになっている。

本システムでは、ICAIに必要な3つのモジュールを持たせたことにより、システム全体としてもかなりまとめることができたと思われる。しかし、そのためにシステム自体が大きなものになり、他のワークステーションへの移植性に多少の課題が残った。

本システムは、富士通のワークステーションFACOM G-150Aを使用し、MACLISP系の言

語の UTILISP (University of Tokyo Interactive LIST Processor)<sup>14,15)</sup> を用いて構築した。

本システムは、1人が1台のワークステーションで使用する環境になっており、それによって、ハードウェアにいっさい手を加えることなくシステムを実行でき、また記憶容量についてもほとんど影響なく使用することができる。しかし、複数の学習者が同時にシステムを使用する場合には、各ワークステーション毎に本システムをインストールする必要がある。また、学生モデルデータが、使用したワークステーションに記憶されるため、学習者毎に使用するワークステーションが固定化される。そこで将来的には、多数の学習者がどのワークステーションからも自由に本システムを利用できるように、ホストコンピュータ FACOM M-380を介する構成に拡張することを計画している。

#### 参考文献

- 1) 和多田作一郎：人工知能を深める本，実務教育出版
- 2) 田村進一・柳原圭雄・唐沢 博：人工知能の世界，技術評論社
- 3) A. Barr and E. A. Feigenbaum 著，田中幸吉・淵 一博 監訳：人工知能ハンドブックⅡ，共立出版
- 4) AI ジャーナル，No. 4，UPU
- 5) G.L. サイモンズ著，粟田照平訳：パソコンエキスパート・システム その開発動向と応用の実際，啓学出版
- 6) 石田晴久：パソコン言語学，アスキー出版
- 7) 知的 CAI 最近の動向，情報処理，Vol. 29，No. 11，1988，情報処理学会
- 8) 上野晴樹：知識工学入門，オーム社
- 9) 黒川利明：LISP 入門，培風館
- 10) 中西正和：Lisp 入門 一第3版一，近代科学社
- 11) ダビット・S. トーレッキー著，小川昭子訳：LISP やさしい記号計算入門，啓学出版
- 12) P. H. ウィンストン・B. K. P. ホーン著，白井良明・安部憲広訳：LISP，培風館
- 13) 山本博資・橋爪正樹：パソコンによる LISP 入門 一増補版一，森北出版株式会社
- 14) FACOM SX/G LISP 説明書 一UTILISP 仕様一，エディション11/30A，富士通
- 15) 名古屋大学大型計算機センターニュース，第66号，Vol. 16，No. 2，1985. 5