Prologプログラミング教育のための知的CAIシステムの開発†

河合和久†† 溝口理一郎††† 願化真志††††
喜納久行††††† 角所収††† 豊田順一†††

プログラミング言語の習得のためには，その言語の文法や基本的なプログラミング技法を覚えていくことと，実際に多くのプログラムを自分自身で作成してみることの二つの学習が必要である．これを計算機による教育システム（CAI）の立場から見てみると，前者は計算機が教師として学生に知識を教授していく対話型教授，後者は学生が道具としての計算機を活用し，自発的に学習していく環境型学習に対応する．こうした観点から，著者らは，対話型教授機能と環境型学習機能を持つPrologプログラミング教育用知的CAIシステムを開発した．このシステムでは，Prologプログラミングに必要な知識を対話等によって教授し，学生が，その知識を用いて自由にプログラミングを行える環境を与える．対話型教授機能は，著者らが開発した論理プログラミングと帰納推論に基づく汎用知的CAIシステム上に，Prologプログラミング教育に必要な専門知識を組み込み実現している．また，環境型学習のためのプログラミング環境としては，プログラムの文法上の誤りを検出，指摘するためのパーサと，プログラムの実行状態を視覚的に表示する視覚的Prologインタプリタを用意している．

1. まえがき

教育のための計算機システムは，計算機のはたす役割に基づいて，二つに分類できる．一つは，計算機が教師の役目をはたし，学生を教育していくもので，CAI研究の初期にみられた電子ページめくり機や，フレーム型CAI，さらに，近年の人工知能の成果を応用した対話型システム等がこの種類に含まれる[1),2)]．他の一つは，学生が計算機を道具として活用し，自発的に学習を進めていくもので，問題解決技能習得のためのプログラミング環境を与えたLOGOプロジェクト[3)]や，計算機上でのゲームやシミュレーションを通して学習を進めるシステム等[2)]がその例である．以下では，前者を対話型教授システム，後者を環境型学習システムと呼ぶ．この二種のシステムは，教育する内容および時期によって使い分けられる．対話型教授は，計算機が学生に知識を教え，学生がそれを獲得していく教材に適し，環境型学習は，学生が思考する環境を整備し，学生自身がそれによって理解を深めていく教材に適している．また，ある教材の学習の初期においては，ゲームやシミュレーション等によって学生の興味を促し，次段階では，対話型教授により必要な知識を与え，最終段階では，環境型学習によって自身の理解をより深いものにしていくのが，計算機を利用した教育の理想的な形といえる．一方，プログラミング言語の学習においても，これと同様のことがなりたつ．最初は，教科書等にあるプログラムを走らせてみることから始め，次に，その言語の文法や基本的なプログラミング技法を覚え，さらに，実際に多くのプログラムを自分自身で作成していくことによって，その言語によるプログラミングを習得していくのが普通である．本システムでは，Prologプログラミング[4)]に必要な知識を対話等によって教授し，さらに，学生がその知識を用いて，自由にプログラミングを行える環境を与えている．対話型教授機能は，著者らが開発した論理プログラミングと帰納推論に基づく汎用知的CAIシステム[5)]上に，Prologプログラミング教育に必要な知識を組み込み実現している．また，環境型学習のためのプログラミング環境としては，プログラムの文法上の誤りを検出，指摘するためのパーサと，プログラムの実行状態を証明木の形式で表示する視覚的Prologインタプリタ（以下，VPI—Visual Prolog Interpreter—と呼ぶ）を用いている．

以下，2章では本システムの教育目標とその課題内容について述べる．3章では対話型教授のための知的CAIシステムと，環境型学習のためのプログラミング環境について述べ，本システムの構成を示す．4章では本システムによる教育例を示す．

† An Intelligent CAI System for Programming in Prolog by KAZUHISA KAWAI (Department of Information and Computer Sciences, Faculty of Engineering, Toyohashi University of Technology), RIICHIRO MIZOGUCHI (The Institute of Scientific and Industrial Research, Osaka University), MASASHI GANKE (Toshiba Corporation), HISAYUKI KINOH (Matsushita Electric Co., Ltd.), OSAMU KAKUSHO and JUN'ICHI TOYODA (The Institute of Scientific and Industrial Research, Osaka University).
†† 豊橋技術科学大学工学部情報工学系
††† 大阪大学産業科学研究所
†††† (株)東芝
††††† 松下電器産業(株)

2. Prolog プログラミング教育

2.1 教育目標

教育対象，すなわち，学生（生徒，ユーザ）としては，Prolog 以外の言語（FORTRAN や Pascal，C 等）の使用経験のある情報工学科の4年次学生を想定している．彼らは，2・3年次の実験や演習等で，いくつかの手続き型言語によるプログラミングの経験があり，TSS 利用やファイル操作の能力もある*．このような学生を，Prolog プログラムの読み書きができるようにするのが，本システムの具体的な教育目標である．このためには，次の三つの項目を習得させなければならない．

1) シンタクスや Prolog の動作，組み込み述語の機能
2) 実際のプログラミングでよく用いられる技法
3) Prolog のベースである論理プログラミングの考え方（パラダイム），特に他言語との相違

2.2 課題内容

本システムの課題内容を**表1**に示す．各課題は，数個から数十個の解説および例題から構成される．

Ⅰ）プログラム例　簡単なプログラム例を示し，Prolog プログラムが事物の関係についての事実と規則からなり，三段論法が用いられ，非決定性動作をとることを説明する．さらに，プログラム例を用いて実際に実行させ，Prolog の感じをつかませる．

Ⅱ）シンタクス　Prolog の構文知識を教授する．Prolog には，種々の処理系があり，構文規則も各処理系によって，少しずつ異なっている．本システムでは，システムの実現言語である MV-Prolog[6] の構文を教授する．

Ⅲ）単一化　Prolog の変数は一度値が代入されると，その値は決して変わらない．この single assignment 性が，他のプログラミング言語と最も異なる点である．さらに，定数同士のマッチング，変数への代入，変数同士の値の共有等について教育する．

Ⅳ）インタプリタ　非決定性動作を実現するために，深さ優先，後戻り制御を行うことと，手続き呼出しが単一化によって行われることを教授し，さらに VPI を用いて，視覚的に説明する．

Ⅴ）組み込み述語　入出力や数値演算，比較，高階制御等のシステム組み込み述語の機能と用法について教育する．

表1 課題内容一覧
Table 1 Problems for student practice.

課　題	概　要
Ⅰ）プログラム例	簡単なプログラムの実行例
Ⅱ）シンタクス	定数，変数，複合項，リスト，オペレータ
Ⅲ）単 一 化	定数同士のマッチング，変数への代入，変数同士の共有化
Ⅳ）インタプリタ	深さ優先，後戻り制御
Ⅴ）組み込み述語	入出力，数値計算，高階述語
Ⅵ）プログラミング技法	再帰，差分リスト，カットの用法
Ⅶ）パラダイム	関係（論理）の記述，非決定性動作

Ⅵ）プログラミング技法　実際のプログラミングでよく用いられる技法を習得することは，プログラム作成の効率化だけでなく，大きなプログラムを理解する際のプログラムの構造理解に有効である．ここで教授する Prolog のプログラミング技法[7]は，リストの再帰，数の再帰，差分リスト等，十数項目で，プログラムを作成させたり，プログラム中の誤りを指摘させたりする実践的な問題を用いて教育する．

Ⅶ）論理プログラミングのパラダイム　本課題では，環境型学習として，実際に学生に多くのプログラムを作成させることによって，論理プログラミングのパラダイムを習得させる．学生は，システムが出題する課題，あるいは，自分自身で用意した問題に対するプログラムを作成し入力する．入力されたプログラムに文法上の誤りがある場合には，システムがその誤りを指摘する．さらに，その誤りに関係する構文規則の説明も用意されている．プログラムは，VPI によって実行され，証明木の形式で表示される．学生は，VPI の表示から，プログラムの動きに対する理解を深めていく．

課題Ⅰ）から課題Ⅵ）は，対話型教授によって教育され，課題Ⅶ）は環境型学習によって学習される．

3. システムの構成

本システムの構成を**図1**に示す．対話型教授においては，専門知識モジュールが専門知識を用いて問題や解説を学生に与え，それに対する学生の応答から帰納推論システム MIS (Model Inference System)[8] が学生モデルを生成する．生成された学生モデルは，バグ

* こうした学生が，著者らの研究室に配属された時に，Prolog の教育を行うシステムを作成しようとしたのが，本研究の直接の動機の一つである．

診断システム PDS (Program Diagnosis System)[8] によって，専門知識と比較され，学生のバグが検出される．個人指導モジュールは，教育戦略知識に基づき，学生のバグに対する次の教育を決定し，それを専門知識モジュール等へ指示する．環境型学習においては，専門知識モジュールによって与えられた問題や，学生自身で用意した問題に対する学生の作ったプログラムが，VPI+パーサに送られ，パーサは専門知識にある構文規則を用いてプログラムの誤り検出を行い，VPIは実行状態を学生に表示する．個人指導モジュールは，このようなシステム全体の動作の制御も行う．

本章では，本システムの各モジュールの詳細について述べる．なお，本システムはスーパミニコン MV/8000 II 上に MV-Prolog を用いて実現している．MV-Prolog は VAX版 C-Prolog を MV/8000 II 上に移植し[6]，さらに種々の機能拡張を施したもので，マルチウィンドウ機能，メニュー機能，グラフィック表示機能を有する．本システムでは，これらの拡張機能を十分に活用し，視覚情報を駆使した分かりやすく扱いやすいシステムの実現を図っている．

3.1 論理プログラミングと帰納推論に基づく汎用知的 CAI

知的 CAI においては，それ以前の CAI が，あらかじめシステム内に用意されたコースウェアに沿って教育を進めるだけで，個々の学生の理解を十分把握できず，とかく画一的な教育になりがちであり，学生にとっても馴染みにくいものであったことから，学生個個の理解をモデル化し，それ以後の教育に反映させる試みがなされている[2]．これには，学生モデル，専門知識，個人指導の三つの機能が必要である．特に学生モデルは，学生の理解，とりわけ，学生の犯した誤りとその原因をモデル化するもので，システムの教育効果を決定するものといえる．学生のバグには，次の三種がある．

1) 知識の欠落
2) 誤った知識
3) 知識の適用の誤り

従来よく用いられてきたモデル化の一つはオーバレイモデル[9]と呼ばれるもので，学生が理解している知識は専門知識の部分集合として記述される．この方法では学生の持つ誤った知識や知識の適用の誤りを表現できないという欠点がある．モデル化のもう一つの方法は，専門知識の混乱状態として前もってシステムが用意している知識を用いるバグモデル[10]である．バグモデルは優れた表現力を持ち，上記の三種のバグをモデル化できるが，その構成法は専門知識の内容に強く依存し，モデル生成の汎用化が困難である．これに対し，著者らは学生モデルと専門知識の表現に論理型言

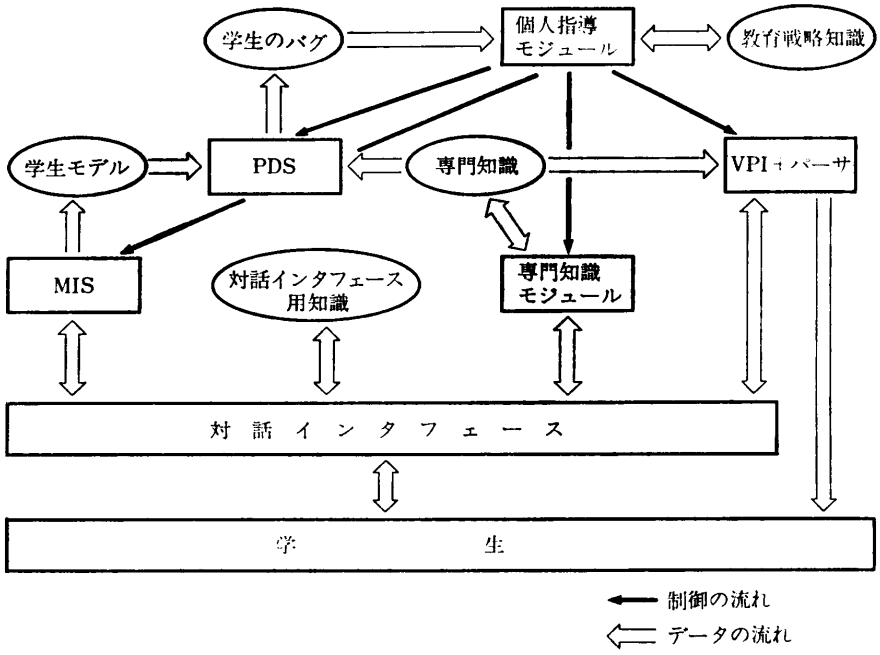

図 1 システムの構成
Fig. 1 Block diagram of our system.

語 Prolog を用い，論理プログラミングのパラダイムによって，知識と知識の適用に関する知識を表現可能にし，その表現上での汎用的な帰納推論を用いて学生モデルを生成することによって，三種のバグのモデル化ができる新しい汎用知的CAIシステムを開発した．汎用システムの詳細については，文献 5) を参照されたい．本システムは，Prolog プログラミング教育システムを構築するために，この汎用システム上に，専門知識と教育戦略知識，対話インタフェース用知識の三種の知識を加え，対話型教授システムを実現し，さらに，環境型学習のための VPI+パーサを統合したものである．

3.2 専門知識と学生モデル

本節では，2.2 節で述べた課題の項目ごとに，専門知識の表現と学生モデルの生成について述べる．

Ⅰ）プログラム例　この課題では，システムが解説を提示していくのみで，学生の応答からの学生モデル生成は行わない．提示される解説は，解説文としてファイルにあらかじめ用意したもので，個人指導モジュールの指示に従い，これを順に提示していく．また，プログラムの実行は，Prolog システムを呼び出すことによって処理する．

Ⅱ）シンタクス　Prolog の構文に関する専門知識は，

```
変数 (i_model, syntax_atomic, 7, X, Z) :-
        英大文字 (i_model, _, _, X, Y),
        文字列 (i_model, _, _, Y, Z).
```

のように，構文要素名を述語とし，構文要素の実体を表す文字列を第 4，5 引数の差分リストとする節によって表現される．節の左辺の第2引数は，その節が含まれる階層世界名である．階層世界は，汎用知的CAIシステムのモデル表現に，メタ知識の取扱いと，表現のモジュラリティ向上のために組み込んだ機能であり[5]，本システムでは，一つの課題に対する専門知識を数個から十数個の階層世界に分けて表現している．節の左辺の第3引数は，節番号で，階層世界名とこの節番号を用いて，この節に関する解説文ファイルや問題文ファイルのファイル名を得る[5]．各述語の第1引数は，その節が専門知識であることを示す．学生モデルの生成は，学生に出題した問題と，それに対する学生の答えから得られる Prolog の事実例集合に対して，MIS を起動することによって行う．得られた事実例集合から MIS によって学生モデルを生成する方法については，文献5) に譲り，ここでは，学生の応答から，いかに事実例集合を得るかについて述べる．本システムの問題は，すべて以下の形式のものである．

（問）　下の選択肢の中で変数として正しいものを選びなさい．

1. var　2. Var　3. xyz　4. 345　5. _

これらの問題は，あらかじめ問題文ファイルに用意されており，各選択肢に対応する事実例が与えられている．例えば，上の例の場合，var に対しては，

変数 (s_model, syntax_atomic, _, [v, a, r], [])

が与えられている*．一方，MIS には，事実例とその事実例が true であるか，false であるかをあわせて送らなければならない．したがって，この事実例に学生の選択に応じて，true, false を与えたものが MIS に送られる．MIS によるモデル生成の途中で，MIS によって出される質問[5]は，次の二種の形式で質問される．

1)　var が変数である理由は．
2)　ar は文字列ですか．

これらは，MIS が行う質問を，対話インタフェースにより変換したものである．

Ⅲ）単一化　単一化に関する専門知識は，変数の値の管理に置換リストを用いて，次のようにまとめられる．

1)　置換リストより第1要素の値を得る
2)　置換リストより第2要素の値を得る
3)　case（第1要素，第2要素）
　　（変数，変数）変数の共有
　　（変数，非変数）代入
　　（定数，定数）=
　　（複合項，複合項）
　　　関数=関数
　　　引数の数の一致
　　　引数同士の単一化

置換リストは，変数とその値の対から成るリストで，Prolog 表現では各要素の表す差分リストの対から成るリストとして表現される．単一化に関する専門知識の Prolog 表現は，4章図6-a）の述語“単一化”の節のようになる．二つの要素の値による場合分けに応じて，四つの節が用いられる．（図6では，最初の二つの節が示されている．）述語“単一化”の第 1〜3 引数は，課題Ⅱ）シンタクスの専門知識と同様，専門知

* この事実例表現は，先に述べた専門知識の場合と同様で，第1引数が学生モデルの節であることを示している．また，第2引数はこの事実例の含まれる階層世界名を，第3引数は節番号（学生モデルの場合，すべて無名変数）を表す．第 4，5 引数は差分リストで，変数である実体 var の文字列を表現している．

識，階層世界名，節番号を示す．第4，5引数は，単一化が行われる第1要素を表す差分リスト，第6，7引数は第2要素を表す差分リストである．第8，9引数は置換リストで，第8引数の置換リストの下で単一化を行い，その結果得られる新しい置換リストが第9引数に返される．学生モデルは，4章図4のICAIウィンドウにみられる問題と，4章図5に示すようなMISからの質問によって生成される．

Ⅳ）インタプリタ　インタプリタに関する専門知識は，ゴールの並び，節の並び，置換リスト，環境の四要素によって構成される．ゴールの並びは，プログラマから与えられるPrologの質問で，充足されるべきゴールの連言である．プログラムは，節の並びとして表現される．節の順序は，プログラマによる入力順である．置換リストは，実行時の変数の値を管理する．インタプリタは，深さ優先，後戻り制御によって，非決定性動作を実現しているので，インタプリタの実行の一ステップごとに後戻りのために必要な情報をスタックにおかなければならない．スタックにおかれる情報は，ゴールの並び，単一化を行った節，置換リストの三つで，これらをまとめて環境と呼ぶ．以上の四要素を用いてインタプリタに関する専門知識は，次のようにまとめられる．

1) ゴールの並びから最左端のゴールを選択する
2) 節の並びから単一化可能な節を選択する
   失敗すれば，6) へ
3) 環境をスタックに保存
4) 新しいゴールの並びの生成
5) 1) へ
6) 古い環境の復活
7) 単一化可能な次候補節を選択する
   失敗すれば，6) へ
8) 3) へ

この専門知識に基づいて，課題Ⅲ）単一化の場合と同様の手法で学生モデルが生成される．

Ⅴ）組み込み述語　この課題では，組み込み述語の機能，用法について教育を行う．現在のシステムでは，組み込み述語の機能に対する知識の統一的なProlog表現が完成しておらず，課題Ⅰ）プログラム例と同様に，解説を提示していくのみで，学生モデルの生成は行っていない．また，組み込み述語の用法についても，一部は，次の課題Ⅵ）プログラミング技法として，専門知識化し，学生モデルの生成も行っているが，大部分については，組み込み述語の機能と同様，解説のみの教育を行っている．これらに対しての検討を加え，専門知識のProlog表現化および学生モデルの生成とそれに基づく教育を行えるようにするのは，今後の研究課題である．

Ⅵ）プログラミング技法　プログラミング技法「[]を停止条件とするリストの再帰」（以下，「この技法」と呼ぶ）を例に，専門知識の内容と，学生モデルの生成について述べる．「この技法」を用いたプログラムについて検討すると，七つの性質を持つことが分かる．

1) 左辺の述語の一つの引数に，ドット対“[変数1|変数2]”が含まれる．
2) 左辺の述語と同じ述語を持つゴールが右辺に含まれる．
3) そのゴールの引数の数は，左辺の述語の引数の数と同一である．
4) そのゴールが，性質1）のドット対の引数位置と同じ位置に引数“変数2”を持つ．
5) 性質2）の述語と同じ述語の事実（停止条件節）がある．
6) その事実の引数の数は，性質3）の引数の数と同一である．
7) その事実は，性質1）のドット対の引数位置と同じ位置に引数“[]”を持つ．

このことより，「この技法」の専門知識は，上記の七つの性質を表すゴールの連言として表現できる．一方，「この技法」の学生モデルは，「この技法」を用いたプログラムを学生に作成させ，そのプログラムが，上の七つの性質を有しているかに従って生成される．上の七つの性質は，すべてプログラムの構文解析によって得られるので，学生が生成したプログラムを専門知識に用意されたDCG規則[11]によって構文解析し，それぞれの性質を持っているかを調べる．検出された性質は事実例の形でMISに送られ，学生モデルが生成される．上の性質のいくつかを持たないプログラムが入力された場合，DCG規則による解析は途中で失敗し，それまでに検出された性質だけがMISに送られる*．

Ⅶ）論理プログラミングのパラダイム　この課題では，VPIを用いて学生に自由にプログラミングを行わせ，環境型学習を進める．出題される問題は，ファイルにあらかじめ用意されたものである．また，学生

* 課題Ⅵ）の学生モデル生成法は一種のオーバレイモデル法になっている．MISによるモデル生成はオーバレイモデルの能力を含んでいる．

自身で用意した任意のプログラムの実行も可能である．パーサは，課題Ⅱ）で述べた Prolog の構文規則に関する専門知識を用いて，プログラムの構文解析を行い，誤りを検出する．VPI+パーサについては，次節で詳しく述べる．

3.3 環境型学習のための VPI

環境型学習の成功例としては，Papert らによる LOGO プロジェクト[3]が最も著名である．LOGO プロジェクトは，10 歳前後の子供に，プログラミングを通して，数学的・論理的問題解決技能の発達を促進させようとするもので，計算機はその技能発達の環境を与えるものである．LOGO プロジェクトが成功した要因としては，①LOGO 言語が子供にとって「無理のない，分かりやすい」言語であること，②構文や用語の誤り等について，あまり細かいことを考えずに進めていける環境，③亀の動きの視覚的なおもしろさ等が挙げられる．これらの要因を考察すると，教育の目標である数学的・論理的問題解決技能の習得という点以外の負担を最小限にしようとした結果，このような視覚的要素を含んだ環境になったといえよう．本システムでは，同様の観点から，論理プログラミングのパラダイムが直感できるような視覚にうったえる Prolog インタプリタを開発し，その上で学生が自由にプログラムを実行することによって，論理プログラミングのパラダイムを習得できる環境を与えている．Prolog の実行を視覚化する方法は，文献 12)，13) 等に提案されているが，本システムでは証明木の生成，消滅の過程を視覚化した新しい Prolog インタプリタ VPI によって実現している．

Prolog の実行において，最も分かりにくいのは，後戻り制御と，単一化による変数の値の決定である．特に，一度にいくつかのゴールにわたって後戻りを行う場合と，共有化した変数のいずれかに値が代入された時に，もう一方の変数にも値が代入されていることが分かりにくい．VPI では，こうした点を特に考慮した表示を行っている．まず，後戻りを含めた Prolog の実行の制御の様子は，証明木の伸張，収縮によって示される．一つのゴールが実行されると，それに対する手続きが表示され，それらの内で単一化の行われた節に印が付けられる（**図 2** ⓐ）．新しいゴール列は，証明木の形式で，最初のゴールの下部に表示される（図 2 ⓑ）．この処理が繰り返され，証明木が完成していく．後戻りが生じると，古いゴールが順番に消去されていき，手続き中の別の節が単一化可能な時には，印がその節の方に付け直され，その下に新しい証明木が表示される．単一化によって変数に値が代入された時には，その時点で表示されている同じ変数がすべてその値に書き替えられる．画面上の表示可能域は，MV-Prolog のグラフィック機能の縮小文字を用いて，縦 60 行，横 120 字である．このため，表示できるプログラムの大きさは，かなり限られたものになってしまっている．これを解決するには，現在のように，プログラムの実行のすべてを均等に表示するのではなく，実行しているゴールの付近だけを現在の大きさで表示し，それ以外の部分は，縮小して表示する等の処理が必要と考えられる．この機能の実現は，今後の課題の一つである．

学生が入力したプログラムは，VPI によって実行される前に，パーサにより専門知識にある Prolog の構文規則に従って解析され，構文上の誤りが検出される．誤りが検出された時には，その誤りを示し，その誤った部分に対応する構文規則の説明を行う．

3.4 個人指導モジュール

本章の初めにも簡単に述べたように，個人指導モジュールはシステム全体の動作の制御を行う．具体的に行う処理は以下の七つである．

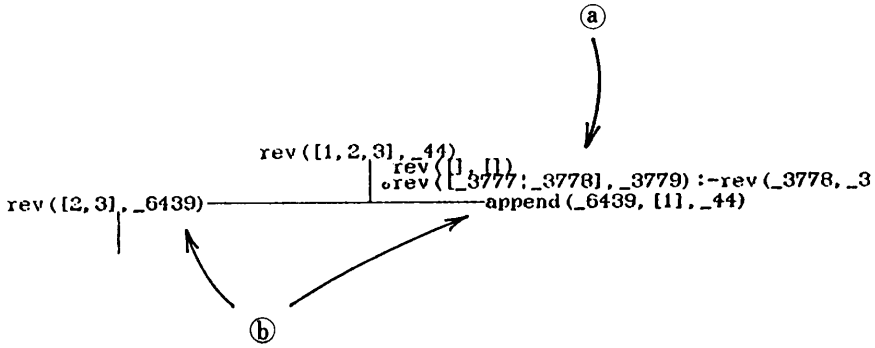

図 2 VPI の表示
Fig. 2 The portion of proof tree displayed by VPI.

1) 解説文の表示を専門知識モジュールに指示する.
2) 問題文の表示を専門知識モジュールに指示し，学生の解答から得られる事実例をMISに送る.
3) MISによるモデル生成を指示する.
4) PDSによるバグ検出を指示する.
5) 学生のバグから次の教育を決定し，専門知識モジュールに指示する.
6) VPI+パーサを起動する.
7) ヒントやメニュー等のコマンドに対する処理を専門知識モジュールに指示する.

上の七つの処理は，モデル生成戦略と再教育戦略の二つの教育戦略知識に従って行われる．モデル生成戦略知識は，処理の実行順を記述したもので，課題ごとに，1) から 5) の処理に対する指示をリスト表現で用意している.

```
[...,
  解説 (file, syntax_term 1),
  問題 (file, syntax_term 1),
  mis,
  pds (file, syntax_term 1),
  再教育,
  ...]
```

これは，モデル生成戦略知識の課題II）シンタクスに関するリストの一部である．“解説 (file, syntax_term 1)” は，ファイル syntax_term 1. epl にある解説文を表示することを指示し，“問題 (file, syntax_term 1)” は同じくファイル syntax_term 1. prb にある問題文を表示し，それに対する学生の解答から，MISに送る事実例を得ることを指示している．“mis” は，得られた事実例に対して，MISを起動することを指示している．MISの中で出される各種の質問については，個人指導モジュールは関与せず，すべての質問が学生に出される．“pds (file, syntax_term 1)” は，ファイル syntax_term 1. pds にある事実例を，学生モデル，専門知識によってそれぞれ実行し，異なる答えが得られるものに対して PDS を起動し，学生モデルに含まれるバグを検出することを PDS に指示する．“再教育” は，PDS によって検出されたバグに対する再教育の実行を指示する．この再教育の方式を記述したのが，再教育戦略知識である．この知識は，バグのあった知識項目とそのバグの種類から，そのバグに対する再教育の方式を与えるもので，各要素に教育方式を値として持つ，知識項目とバグの種類を各々縦軸，横軸とするマトリクスとして表現される．教育方式としては，

1) 正答提示方式 学生のバグに対応する専門知識内の項目，すなわち，誤りに対する正答を提示する.
2) 例題提示方式 学生のバグが誤答の唯一の原因となる問題を次々に学生に与え，学生自身にその誤答の原因に気付かせる.
3) ソクラテス教育方式 学生がバグを持っている概念について，その下位概念の教育に用いられる方式で，複数個の下位概念のうち，いくつかを欠落した場合を次々に学生に提示し，そうした誤り例を通して，学生に正しい概念を教授していく.

表 2 コマンド一覧
Table 2 Commands and their functions.

コマンド	機能
next	次の解説または問題に進む
QA	解答あるいは質問の入力
skip	次の世界に進む
back	直前の世界に戻る
bend	以前の世界に復帰する
menu	全課程から次の世界を選択する
hint	問題のヒントが表示される
detail	より詳細な解説が表示される
ans	問題の解答が表示される
help	コマンドの説明
prolog	Prolog ゴールの実行
end	終了

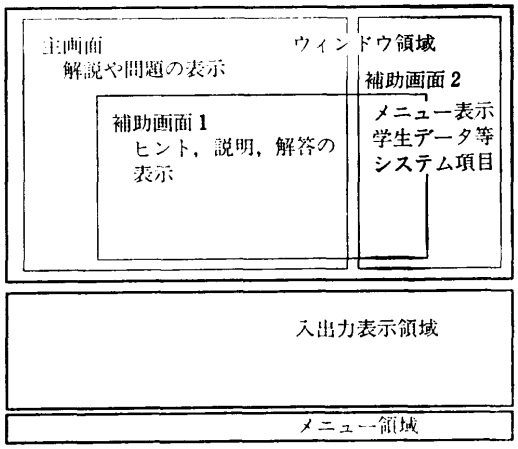

図 3 画面表示の割付け
Fig. 3 Screen design.

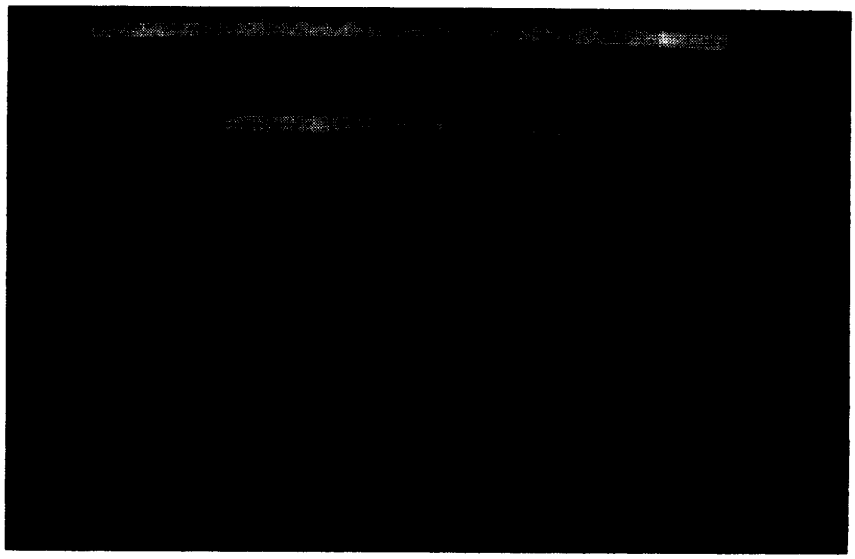

図 4 単一化に関する教育例
Fig. 4 Example of a training session on unification.

```
1 と X が単一化可能な理由は.
  理由は文節単位に単文で入力してください.
  最後には＊＊＊を入力してください.
》》》 1の値は1である.
》》》 X の値は1である.
》》》 1と1は等しい.
》》》 ＊＊＊
2 と Y が単一化可能な理由は.
  理由は文節単位に単文で入力してください.
  最後には＊＊＊を入力してください.
》》》 2の値は2である.
》》》 Y の値は X である.
》》》 X に2を代入する.
》》》 ＊＊＊
```

図 5 MIS による質問例
Fig. 5 Query examples by MIS.

の三種がある.

処理 6) の VPI+パーサの起動は，課題 Ⅶ) パラダイムの教育において行われる．また，処理 7) の各種コマンドに対する処理は，次節の対話インタフェース機能で述べるように，常に学生の入力を監視し，コマンドの入力された時点で実行される．

3.5 対話インタフェース

対話インタフェースは，システムの使い心地を決定する最も重要な機能と考えられ，種々の研究が進められている．本システムでは，MV-Prolog の持つ各種機能を活用し，扱いやすいシステムを実現している．システムの用意しているコマンドは，**表 2** のとおりで，入力待ちの時はいつでも，学生から質問を行ったり，あるいは他の課題へ移行したり，終了したりでき

```
単一化(i_model,unify,1,X1,X2,Y1,Y2,S,S0) :-
        値の取出し(i_model,_,_,X1,X2,S,XV1,XV2),
        値の取出し(i_model,_,_,Y1,Y2,S,YV1,YV2),
        変数(i_model,_,_,XV1,XV2),
        変数(i_model,_,_,YV1,YV2),
        共有(i_model,_,_,XV1,XV2,YV1,YV2,S,S0).
単一化(i_model,unify,2,X1,X2,Y1,Y2,S,S0) :-
        値の取出し(i_model,_,_,X1,X2,S,XV1,XV2),
        値の取出し(i_model,_,_,Y1,Y2,S,YV1,YV2),
        変数(i_model,_,_,XV1,XV2),
        非変数(i_model,_,_,YV1,YV2),
        代入(i_model,_,_,XV1,XV2,YV1,YV2,S,S0).
                ⋮
値の取出し(i_model,unify,11,X1,X2,_,X1,X2) :-
        非変数(i_model,_,_,X1,X2).
値の取出し(i_model,unify,12,X1,X2,S,X1,X2) :-
        変数(i_model,_,_,X1,X2),
        not(member(((X1,X2) $ _),S)).
値の取出し(i_model,unify,13,X1,X2,S,Z1,Z2) :-
        変数(i_model,_,_,X1,X2),
        member(((X1,X2) $ (Y1,Y2)),S),
        値の取出し(i_model,_,_,Y1,Y2,S,Z1,Z2).      ---ⓐ
```

a) 専門知識

```
単一化(s_model,unify,_,X1,X2,Y1,Y2,S,S0) :-
        値の取出し(s_model,_,_,X1,X2,S,XV1,XV2),
        値の取出し(s_model,_,_,Y1,Y2,S,YV1,YV2),
        変数(s_model,_,_,XV1,XV2),
        変数(s_model,_,_,YV1,YV2),
        共有(s_model,_,_,XV1,XV2,YV1,YV2,S,S0).
単一化(s_model,unify,_,X1,X2,Y1,Y2,S,S0) :-
        値の取出し(s_model,_,_,X1,X2,S,XV1,XV2),
        値の取出し(s_model,_,_,Y1,Y2,S,YV1,YV2),
        変数(s_model,_,_,XV1,XV2),
        非変数(s_model,_,_,YV1,YV2),
        代入(s_model,_,_,XV1,XV2,YV1,YV2,S,S0).
                ⋮
値の取出し(s_model,unify,_,X1,X2,_,X1,X2) :-
        非変数(s_model,_,_,X1,X2).
値の取出し(s_model,unify,_,X1,X2,S,X1,X2) :-
        変数(s_model,_,_,X1,X2),
        not(member(((X1,X2) $ _),S)).
値の取出し(s_model,unify,_,X1,X2,S,Y1,Y2) :-
        変数(s_model,_,_,X1,X2),
        member(((X1,X2) $ (Y1,Y2)),S).              ---ⓐ
```

b) 学生モデル

図 6 単一化に関する知識の Prolog 表現
Fig. 6 Knowledge of unification in Prolog.

る．これらのコマンドは，MV-Prolog のメニュー機能によって，画面の最下部に常に表示され，カーソルキーによって選択される．学生への問題や解説，ヒント等は，マルチウィンドウ機能を用いて，異なるウィンドウに表示される．**図3**に，こうした画面上の各領域の割付けを示す．

対話インタフェースは，上記の表示機能のほかに，Prolog による内部表現と，自然言語表現との変換を行う[5]．変換は，簡単なテンプレートマッチングによって行っており，変換のための辞書は，Prolog で書かれた知識として保持している．

4. 教 育 例

図4は，課題Ⅲ）単一化に関する問題において，学生がヒントを要求した際の画面である．画面左上の ICAI ウィンドウに表示されている単一化可能な組を選択する問題に対して，画面中央の HINT ウィンドウには単一化の手順に関する解説文が表示されている．この問題に対して学生が 1, 2 を単一化可能として選択した場合（正答は，1, 3），MIS から**図5**のよ

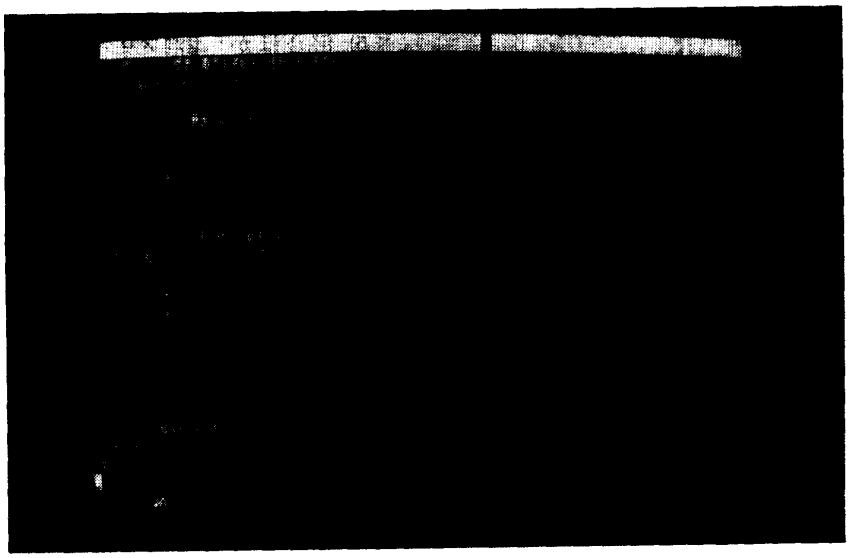

図7 プログラミング技法に関する教育例
Fig. 7 Example of a training session on programming technique.

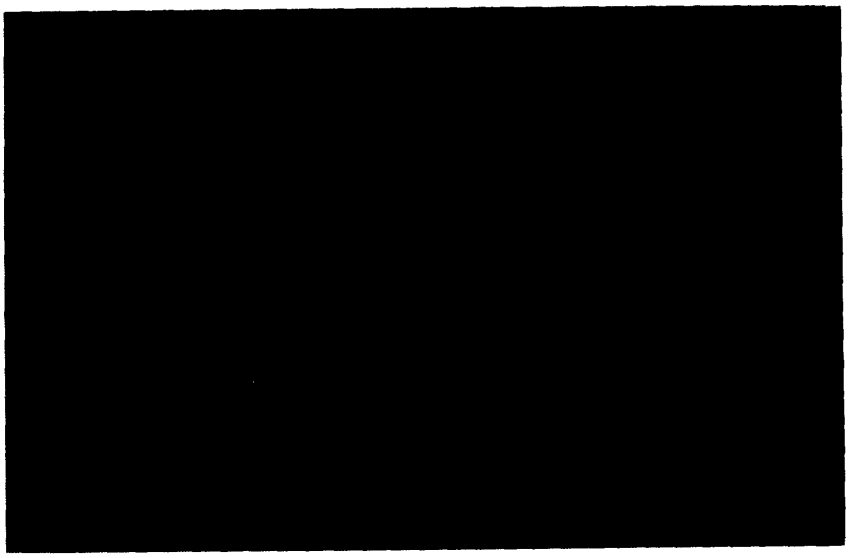

図8 VPI の出力例
Fig. 8 Example of VPI's output.

うな質問が出された時点で，図6-b）のような学生モデルが生成される．この学生は，図5の理由を述べる質問において，値の取出しに関して，Y の値は X, X の値は1であるとそれぞれ答えていることから，変数の値の取出しにおいて，その値が変数であった場合，さらにその変数の値をもう一度取り出さなければならないことを知らないと思われる．学生モデルを図6-a）の専門知識と比較すると，学生が値の取出しにおいて，取り出した値が変数である場合に，さらに値の取出しを繰り返していないことがモデル化されていることが分かる（図中ⓐ）．

図7は，課題Ⅵ）プログラミング技法のリストの再帰に関する問題において，学生が解答したプログラムを解析し，その誤りの個所を指摘している画面である．システムは，リストの再帰に関する構文規則によって，学生のプログラムを解析し，第二節の左辺の [] が誤っていることを検出，指摘している．

図8は VPI の表示例で，rev ([1, 2, 3], X) に対する解が求まった状態を示している．

5. むすび

プログラミング言語の習得における教授と学習の重要性に基づき著者らが開発した，対話型教授機能と環境型学習機能を持つ Prolog プログラミング教育用知的 CAI システムについて述べた．本システムの特徴をまとめると，次のようになる．

1) 論理プログラミングと帰納推論に基づく汎用知的 CAI システムによる正確な学生モデル生成機能を持つ対話型教授
2) VPI による視覚的な環境型学習
3) マルチウィンドウ，メニュー，グラフィック表示機能を駆使した分かりやすい対話インタフェース

今後，システムの教育効果を評価，検討し，システムの問題点を抽出，修正していくことが課題としてあげられる．

謝辞 日頃，熱心な御討論，有益な御助言をいただく大阪大学産業科学研究所上原邦昭助手並びに山口高平助手，中村孝技官に深く感謝します．また，MV-Prolog の拡張および本システムの開発に際し，種々の御協力をいただいた大阪大学大学院生中村祐一氏，林浩一氏（現在，富士ゼロックス），藤井邦和氏（現在，日本 IBM）に深謝します．

参考文献

1) Carbonell, J. R.: AI in CAI: An Artificial Intelligence Approach to Computer-aided Instruction, *IEEE Trans. Man-Mach. Syst.*, Vol. MMS-11, No. 4, pp. 190-202 (1970).
2) Barr, A. and Feigenbaum, E. A.: *The Handbook of Artificial Intelligence*, Vol. II, pp. 225-235, Pitman, London (1983).
3) パパート, S.: マインドストーム，未来社，東京 (1982).
4) Clocksin, W. F. and Mellish, C. S.: *Programming in PROLOG*, Springer-Verlag, New York (1981).
5) 河合ほか：論理プログラミングと帰納推論による汎用知的 CAI システム，情報処理学会論文誌，Vol. 26, No. 6, pp. 1089-1096 (1985).
6) 藤井ほか：C-Prolog のスーパーミニコン MV/8000 Ⅱ への移植，第28回情報処理学会全国大会，2G-7, pp. 1035-1036 (1984).
7) 中村ほか：Prolog プログラミング教育におけるプログラム例を用いた対話に関する検討，第30回情報処理学会全国大会，4L-3, pp. 1439-1440 (1985).
8) Shapiro, E. Y.: *Algorithmic Program Debugging*, MIT Press, London (1982).
9) Goldstein, I. P.: The Genetic Graph: A Representation for the Evolution of Procedural Knowledge, in Sleeman, D. et al. (ed.), *Intelligent Tutoring Systems*, pp. 51-77, Academic Press, London (1982).
10) Brown, J. S. and Burton, R. R.: Diagnostic Models for Procedural Bugs in Basic Mathematical Skills, *Cognitive Science*, Vol. 2, pp. 155-192 (1978).
11) Pereira, F. C. N. and Warren, D. H. D.: Definite Clause Grammars for Language Analysis, *Artif. Intell.*, Vol. 13, pp. 231-278 (1980).
12) 後藤：Prolog の図形的な動作表示法，情報処理学会記号処理研究会資料，27-6 (1984).
13) 沼尾，藤崎：Prolog プログラミング環境の作成，情報処理学会オペレーティングシステム研究会資料，27-3 (1985).

(昭和60年10月25日受付)
(昭和61年12月10日採録)

河合 和久（正会員）

昭和33年生．昭和61年大阪大学大学院基礎工学研究科博士課程修了．同年豊橋技術科学大学工学部助手．工学博士．知的 CAI，日本語入力，創造工学等の研究に従事．電子情報通信学会，日本認知科学会，CAI 学会，人工知能学会，AAAI，ALP 各会員．

溝口理一郎（正会員）

昭和23年生．昭和47年大阪大学基礎工学部電気工学科卒業．昭和52年同大学院博士課程修了．同年大阪電気通信大学講師，昭和53年大阪大学産業科学研究所助手．現在同研究所助教授．工学博士．パターン識別関数の学習，クラスタ解析，音声の分析・認識・理解，データベースの開発とその知的マン・マシンインタフェース，エキスパートシステム，知的 CAI の研究に従事．1983年度 Pattern Recognition Society 論文賞受賞．電子情報通信学会，日本音響学会，日本認知科学会，CAI 学会，IEEE 各会員．

願化 真志（正会員）

昭和35年生．昭和58年大阪大学基礎工学部情報工学科卒業．昭和60年同大学大学院基礎工学研究科博士前期課程修了．同年東芝(株)入社．現在同社日野工場通信システム技術第二部技術第四担当所属．人工知能に興味を持つ．

喜納 久行（正会員）

昭和34年生．昭和58年大阪大学工学部電子工学科卒業．昭和60年同大学院工学研究科前期課程修了．同年，松下電器産業(株)入社．現在，同社無線研究所勤務．在学中，帰納推論と学習および知的 CAI の研究に従事．電子情報通信学会会員．

角所 収（正会員）

大正15年生．昭和25年大阪大学工学部通信工学科卒業．昭和32年大阪大学産業科学研究所勤務．現在，同研究所教授．工学博士．超音波，電子応用計測，医用電子装置，音声パターン認識，心理音響，ネットワーク理論，信号処理，および知的情報処理システムに関する研究に従事．1983年度 Pattern Recognition Society 論文賞受賞．日本音響学会，電子情報通信学会各会員．

豊田 順一（正会員）

昭和13年生．昭和36年大阪大学工学部通信工学科卒業．昭和41年同大学院博士後期課程退学．同年大阪大学基礎工学部助手．昭和43年工学博士．昭和44年助教授．昭和57年大阪大学産業科学研究所教授．現在，主として自然言語理解および文書画像処理と感性等の研究に従事．人工知能学会，電子情報通信学会，日本認知科学会各会員．