

## 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†

河合和久<sup>††</sup> 溝口理一郎<sup>†††</sup> 喜納久行<sup>††††</sup>  
願化真志<sup>††\*\*</sup> 角所収<sup>†††</sup> 豊田順一<sup>†††</sup>

知的 CAI システムは、専門知識、学生モデル、個人指導の三つのモジュールから構成される。このうち、学生モデルは学生の理解を示すもので、このモデルがいかに実際の学生の理解と一致しているかが、知的 CAI システムの教育効果を決定するといえる。本論文では、こうした観点から、従来の知的 CAI システムにおけるモデル生成法の欠点を指摘し、帰納推論に基づく新しいモデル生成法を持つ汎用知的 CAI システムについて述べる。このシステムでは、学生モデルの生成を、学生の多くの応答事例から、その背後にある学生の理解を推測する帰納手続きとみなし、汎用の帰納推論アルゴリズムを用いることによって、教材に依存しないモデル生成機構を実現している。さらに、モデル表現に論理プログラミングパラダイムを取り入れ、教材や学生の理解に含まれるメタ知識を扱えるようにしている。また、本汎用知的 CAI システムを用いた化学反応教育システムの実現を通して、システムの有用性を示す。

### 1. ま え が き

SCHOLAR<sup>1)</sup>の開発以来、CAI (Computer-Assisted Instruction) 研究に、人工知能や知識工学の研究成果が応用されるようになってきている<sup>2)</sup>。これらはそれ以前の CAI と区別され知的 CAI と呼ばれる。CAI システムは、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あり、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、学生にとっても馴染みにくいという問題があった。これらのことへの反省から、知的 CAI システムでは、学生個々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。本論文では、学生の理解をモデル化するための従来のモデル生成法について考察し、その欠点を補う方法として、帰納推論<sup>3)</sup>を用いることにより、高い記述力と汎用性を持つ新しい汎用知的 CAI システムが実現できることを示す。さらに、モデル表現に論理型言語 Prolog<sup>4)</sup>

を用い、論理プログラミングパラダイムに基づくことによって、教材や学生の理解に含まれるメタ知識を容易に取り扱えることを示す。

以下、2章では知的 CAI システムの構成と従来のシステムの学生モデル生成法の問題点について考察する。3章では論理プログラミングに基づくモデル表現と、帰納推論によるモデル生成について述べ、汎用知的 CAI システムの構成を示す。4章では応用例として化学反応教育システムについて述べる。

### 2. 従来の知的 CAI システムとその問題点

一般に、知的 CAI システムは、図1に示すような構成を持つと考えられている<sup>2)</sup>。

#### 1) 専門知識モジュール

学生に教授すべき分野に関する知識を持ち、学生への問題や解説の生成、学生の答えの評価、並びに、学生からの質問への解答等を行う。

#### 2) 学生モデルモジュール

各学生について、何をどこまで理解したか、また、どの部分がわかっていないかを表すモデルを構成し、そのモデルを蓄積する。

#### 3) 個人指導モジュール

学生モデルに基づいて次にどのような解説や問題を与えるかを決定し、専門知識モジュールに指示する。さらに、システム全体の制御を行う。

上記の三つのモジュールは、知的 CAI システムの構成上、いずれも必要不可欠なものであるが、特に学生モデルモジュールが、学生の理解を十分把握できな

† A Framework for Intelligent CAI Systems Based on Logic Programming and Inductive Inference by KAZUHISA KAWAI (Faculty of Engineering Science, Osaka University), RIIICHIRO MIZOGUCHI (The Institute of Scientific and Industrial Research, Osaka University), HISAYUKI KINOH (Faculty of Engineering, Osaka University), MASASHI GANKE (Faculty of Engineering Science, Osaka University), OSAMU KAKUSHO and JUN'ICHI TOYODA (The Institute of Scientific and Industrial Research, Osaka University).

†† 大阪大学基礎工学部情報工学科

††† 大阪大学産業科学研究所

†††† 大阪大学工学部電子工学科

\* 現在 松下電器産業(株)

\*\* 現在 東芝(株)

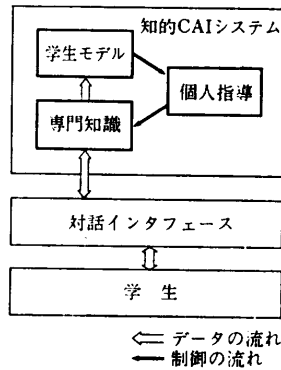


図1 知的 CAI システムの構成

Fig. 1 Block diagram of general ICAI systems.

ければ、学生に適切な問題や解説が与えられず、教育効果が十分に上がらない。すなわち、学生の理解、とりわけ、学生の犯した誤りとその原因をどれだけ正確にモデル化できるかがシステムの教育効果を決定するといえる。したがって、学生モデルの表現形式と生成手法がシステム構成上の最も重要な点となる。

学生の解答に見られる誤りには、答えを導けないもの、誤った答えを導くもの等があるが、それらの原因となる学生の持つバグ（思い違い）は、次の三種類に分類できる。

- 1) 知識の欠落：教授している知識そのものが欠落している場合で、例えば、 $\text{AgCl}$  が沈殿することを知らない等である。
- 2) 誤った知識：誤った知識を持っている場合で、例えば、 $\text{AgNO}_3$  が沈殿すると思いついでいる等である。
- 3) 知識の適用の誤り：正しい知識を持っているが、その使い方が誤っている。

従来の知的 CAI システムでは、二種のモデル化が用いられてきた。一つはオーバレイモデル<sup>5)</sup>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<sup>6)</sup>である。バグモデルは優れた表現力を持つが、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、著者らは論理プログラミングと帰納推論を用いた新しい汎用知的 CAI システムを開発した<sup>12)</sup>。このシステムでは、学生モデルと専門知識の表現に論理型言語 Prolog を用い、論理プログラミングのパラダイムによって、知識の適用に関する知識を表

現可能にしている。さらに、この表現上での帰納推論を用いて学生モデルを生成することにより、上記の三種のバグも含めてモデル化可能にしている。

### 3. 汎用知的 CAI システムの構成

著者らの開発した汎用知的 CAI システム全体の構成を図2に示す。帰納推論アルゴリズムとしてモデル推論システム MIS (Model Inference System)<sup>7)</sup>を用い、専門知識モジュールによって与えられる問題に対する学生の応答から、学生の理解のモデルである Prolog プログラムを生成する。学生の犯す誤りの原因(バグ)は、プログラム診断システム PDS (Program Diagnosis System)<sup>7)</sup>によって、Prolog プログラムのバグとして検出される。検出されたバグは個人指導モジュールに送られ、以後の教育に活用される。

汎用とは、学生モデルの生成法が知的 CAI システムの適用領域、すなわち、教育する分野から独立していることを指し、この汎用知的 CAI システムに専門知識と教育戦略知識、および、対話インタフェース用の知識を加えることによって、種々の専用教育システムが構築できることを意味している。

以下に、本システムの各モジュールの詳細について述べる。

#### 3.1 論理プログラミングによるモデル表現

##### 3.1.1 論理型言語 Prolog

知的 CAI システムにおける専門知識や学生モデル

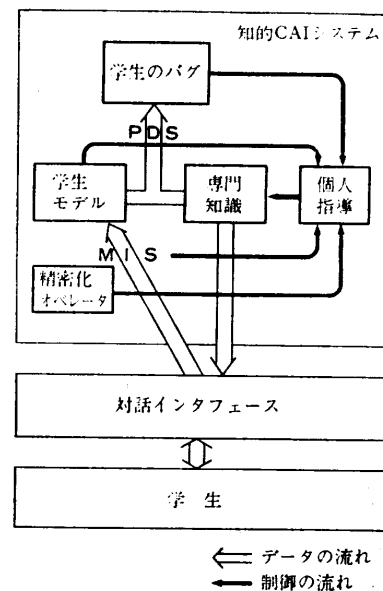


図2 汎用知的 CAI システムの構成

Fig. 2 Diagram of our framework for ICAI systems.

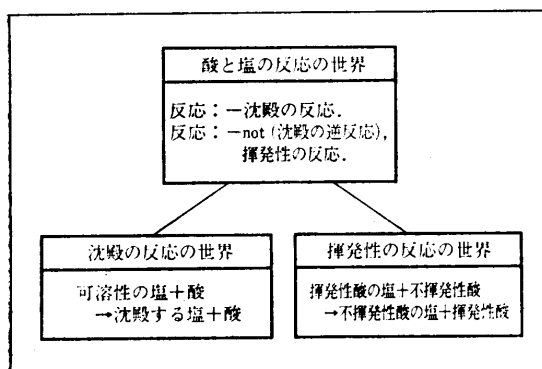
は、教材に関する単なるデータベースではなく、教材間の因果関係に関する推論や、問題解決、演繹等の手続き的知識を含んだものである。これに対し、論理型言語 Prolog はプログラムの宣言的解釈と手続き的解釈が可能で、データとそれに対する推論や手続きを同一の表現内に取り込める。このことより、本システムのモデル表現言語として、Prolog を採用する。

### 3.1.2 メタ知識

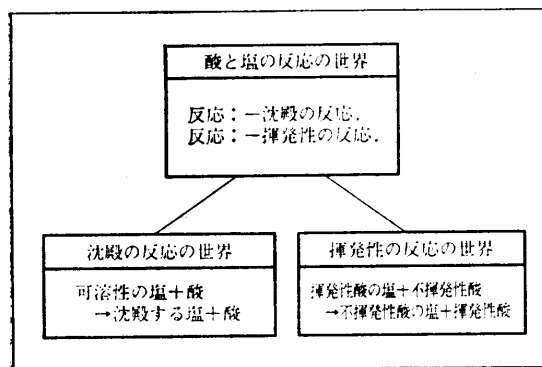
基本問題ならば解けるが、応用問題になると解けない。すなわち、個々の知識は理解しているが、それらをうまく使いこなせないという学生がよくいる。これは、前章で述べたバグのうち、知識の適用の誤りのバグを学生が持っている場合である。こうした学生の理解をモデル化するには、知識の適用に関する知識、すなわち、メタ知識が扱えなければならない。例として、塩の沈殿と酸の揮発性に関する酸と塩の反応を考える。

#### (1) 可溶性の塩+酸

→ 沈殿する塩+酸



(a) 専門知識



(b) 学生モデル

図 3 酸と塩の反応に関する知識

Fig. 3 Schematic representation of knowledge for reaction of acids and salts.

#### (2) 揮発性酸の塩+不揮発性酸

→ 不揮発性酸の塩+揮発性酸

上の二つの知識は、各々、沈殿の概念を用いて反応を考える問題と、揮発性の概念を用いて反応を考える問題に用いられる基本的な知識である。ところが、この二つの知識をまとめて適用しなければならない応用問題においては、二つの知識の適用の順序を知らなければならない。すなわち、与えられた問題において、まず沈殿する塩が生成されるかを調べ、生成されるならば、(1)に従い反応が進む。(1)の条件が満たされない場合、(2)の知識を適用するが、このとき、与えられた問題の逆方向の反応が(1)の条件を満たすならば、その逆方向の反応が進むことになる。以上を模式的に書くと、図 3 (a) のようになる。これに対し、世界(1)、(2)の基本的な問題は解けるが、応用問題になると、まず沈殿を考え、次に揮発性を考えてしまい、逆反応が進むかどうかを考慮しないような学生のモデルは図 3 (b) のように表現される。

本システムでは、このようなメタ知識を図 3 のように階層世界を用いて表現する。図 3 では、酸と塩の反応の世界が下位の二つの世界のメタ知識となっている。そして、学生の持つ知識の適用の誤りのバグは、図 3 (b) のようにメタ知識の世界内のバグとして表現され、他の二種のバグと区別される。さらに、階層世界表現は、メタ知識のみでなく、章・節等の形で表現される教材自身の持つ階層性の表現にも適用できる(図 6 参照)。こうした階層性によって、専門知識の構成にモジュラリティを与え、柔軟な専門知識作製が可能となる。学生モデル生成においては、モデル生成の対象を一つの世界に限定することによって、モデル生成の効率化が図れる。本システムでは、Prolog 表現中の全述語の一つの引数にその述語が含まれる世界名を記述することによって階層世界を実現している。

上記のようなメタ知識表現においては、どれをメタ知識とし、どれを通常の知識とするかは、相対的なものであり、教材を作製する者の考え方に依存する。重要な点は本システムが階層世界表現機能を持ち、教材作製者の意図するメタ知識をモデル表現できることである。

### 3.2 帰納推論によるモデル生成

学生モデルは学生の理解度を表現するものであり、その生成は学生の応答や質問という具体例から学生の理解の全体像を帰納する手続きである。したがって、学生モデル表現言語上の(論理プログラミングパラダ

イムにおける) 一般的な帰納推論を用いれば, モデル化するすべての学生の知識が生成可能である. このとき, 生成されたモデルが専門知識からみて, 誤りを含んでいるか否かは無関係であり, 前章に述べた三種のバグもすべてモデル化できる.

### 3.2.1 MIS とモデル生成

本システムの学生モデル表現は, 既に述べたように Prolog を用いており, モデル生成のための帰納推論としては, モデル推論アルゴリズムの一つである MIS を用いている. MIS は, 次々に与えられる Prolog の事実例に対して, それらを証明する Prolog プログラムを生成するもので, プログラム生成の過程は, 仮説の設定と検証を繰り返し, すべての事実例を証明できる仮説を作り上げていくものである. CAI においては, 事実例は学生への問題とそれに対する学生の応答の対によって与えられる. 仮説を設定する際に, MIS は次の三種の質問を oracle (神託, 生成している Prolog プログラムに関するすべてのことを知っている者)<sup>7)</sup> に行う. ①仮説として設定される新しい節の右辺の述語群. ②各述語の引数の構造. ③右辺の各述語の真理値. MIS はこれらの質問に対する oracle の解答から新しい節を生成する. この節生成機能を持つのが精密化オペレータ<sup>7)</sup>である. 本システムでは, 最も一般的な精密化オペレータを用意し, 理論的に可能なすべての節を生成できるようにしている. 仮説が設定されると, 与えられた事実例の証明が可能か検証する. この検証において, 証明不可能な事実例がある場合, その事実例の証明過程のどこにバグがあるかを発見する際にも oracle に質問が出される. この質問は, ④証明過程の各ゴールの真理値についてである. このバグ発見機能を持つのが PDS である. 発見されたバグはプログラムから取り除かれ, 次の仮説が立て直される. 学生モデル生成において, oracle となるのは, 生成しているモデルに関するすべてのことを知っている者, すなわち, 学生自身である.

### 3.2.2 MIS によるモデル生成の評価

MIS による学生モデル生成にはいくつかの問題点がある. この項では, それらを個々に考察し, MIS によるモデル生成の正当性を示す.

まず, 生成可能なモデルの範囲の問題がある. MIS におけるモデル生成の範囲は, 精密化オペレータの能力によって決定される. 本システムでは, 前項で述べたように最も一般的な精密化オペレータを用意しているので, oracle によって指示されるすべての節を生成

できる. しかしながら, 前項④の質問に対して, oracle がシステムに用意していない述語を答えた場合, その節の意味付けができない. 例えば, システム内に沈殿や難溶性等の述語が用意されているときに, 学生が deposit (『沈殿』) という述語を用いた場合, システムは述語 deposit を用いたモデルを生成するが, それが述語沈殿と同じ意味とは知らないで, 学生のバグとしてしまう. このように, 本システムでは, システムに用意している述語を用いて表現できる節以外のものを生成しても教育上意味がない. このことより, 現在のシステムでは, システムに用意している述語を用いて表現できるモデルのみを生成している. この場合, oracle への④の質問は必要ないが, モデル生成の効率化, および, 以下で述べる教育効果を考え, 学生に質問している. また, このモデル生成がバグモデルと等価な記述能力を有していることは明らかである. さらに, バグモデルでは生成するモデルの知識項目(本システムの節にあたる)をすべて用意しておかなければならないが, 本システムでは述語を用意しておくだけで, それらの組合せによって得られるすべての節を生成できる.

次に, モデル生成の効率を考える. MIS のモデル生成の効率化は, 与えられる事実例と精密化オペレータに依存する. モデル生成の効率化には, 実際の学生の理解と学生モデルが異なっている部分の問題を与えることが有効である. 一般に, 学生の理解は専門知識によって表されるモデルに近いと考えられる. したがって, 本システムではモデル生成のための出題の際に, 学生モデルと専門知識によってその問題を実行し, 異なった答えを出すものを優先的に出題するようにしている. また, 現在のシステムでは最も一般的な精密化オペレータを用いているが, これをより個別的なものにすればモデル生成の効率化が図れる. 本システム上に, 専用教育システムを開発する際に, 学生の犯しやすいバグを含むモデルの生成を優先する等の, 教材に依存した精密化オペレータを用いることが有効である.

第三の問題は, MIS によるモデル生成の終了の判定である. 一般に, 学生モデルの生成の終了を正確に知ることは不可能である. これは人間の教師においても同様である. 本システムでは, あらかじめ定められた問題を出した後, 一定期間学生モデルに変化がない場合にモデル生成が終了したものとみなしている.

第四の問題は, oracle への四種の質問の処理である. これらの質問が教育においてどのような意味を持

```

①Pb(NH(3))(2) + H(2)CO(3) -> PbCO(3) + 2NH(3)
:: DO_
上のように答えた理由は?
理由は次の例のように文節単位に単文で答えて下さい。
例 HClは酸である。
AgClは沈殿する。
(最後に *** を入力して下さい。)
:: Pb(NH(3))(2)は塩でない。 ----- ㉔
:: ***
PbCO(3)は沈殿しますか? yes. ----- ㉕

```

図 4 学生への質問例

Fig. 4 Examples of queries set to student.

つかを考えなければならない。四種の質問のうち、②の各述語の引数の構造に関しては、学生の知識と Prolog 表現との対応付けにあたるもので、あらかじめシステム内に用意されており、学生には質問されない。また、④の質問は本質的に③の質問と同じである。したがって、残りの二種の質問が問題となる。それらが教育的にみて意味のある質問形式に直して出題可能であり、かつ教育効果を持つこと、さらに質問が行われる時期も教育的に意味のある時点であることを化学反応教育を例に以下に示す。

1) 節の右辺の述語群：右辺の述語群は、左辺の述語が表す概念の推論過程や理由等を表現する。したがって、図 4 の㉔のように、理由を尋ねる等の質問として学生に与えられる。質問される時期は、モデル生成に新しい述語が必要となったときで、学生が新しい概念を用いた直後である。学生にとっては、自分の思考過程の確認となる。

2) 右辺の各述語の真理値（証明過程の各ゴールの真理値）：MIS が取り扱っている事実例の証明過程に表れる概念の真理値を尋ねるもので、図 4 の㉕のような形式で質問される。これは、現在注目している概念の下位概念に関する基礎演習的な問題といえる。

最後に、学生の理解の変化の問題があげられる。MIS では、与えられる事実例や oracle の応答には、矛盾がないことが前提となっている。この前提は、本システムに関していえば、学生モデルの生成中は、学生の理解に変化があってはならないことを意味する。ところが、モデル生成過程での種々のやりとりで、学生が自身の誤りに気付く、それを改めることは十分考えられる。この問題に対し、本システムでは、学生が自身の考えを改めた場合は、システムからの任意の質問に対してまずそのことを指示し、その後で質問に答えるようにさせている。システム側は、学生から考えの変更の指示があったとき、それ以前の応答事例のうち、学生が変更した考えが関係するものを抽出し、そ

れらを除去した上で、モデルを生成し直す。

### 3.2.3 メタ知識とモデル生成

学生モデルは、専門知識同様、階層世界化されている。この階層性はメタ知識を表現するためのものであるから、モデル生成においては、その時点で生成を行っている世界よりも下位にある世界の基本的な知識の教育は終わっており、専門知識とほとんど同一の学生モデルができていると考えられる。したがって、下位の世界内で学生が誤りを犯すとは考えにくい。このことより、学生モデル生成のために、MIS が取り扱い質問等を出す範囲をその時点の世界に限定しモデル生成の効率化を図る。そして、その世界内ではモデル化できない場合のみ、下位世界のモデルを生成し直す。したがって、メタ知識の教育を行っているときにはメタ知識の世界のみでモデル化が行われ、学生がメタ知識のバグすなわち知識の適用の誤りを持つ場合には、そのバグが MIS によってモデル化される。

### 3.3 PDS によるバグ検出

PDS は、誤った実行結果を返す事実例と Prolog プログラムに対して、その誤りの原因となるバグを、oracle に質問しながら発見するアルゴリズムである。学生のバグ発見においては、PDS に与えられるバグを含んだプログラムは学生モデルであり、oracle は専門知識である。したがって、oracle への質問が学生に出されることはない。発見されるバグは、Prolog レベルで表現すると、1) 節がない、2) 誤った節があるの二種である。これらは、それぞれ 2 章で述べた知識の欠落、誤った知識のバグに対応する。また、メタ知識を表す世界に含まれるバグは、知識の適用の誤りに対応する。このように、PDS を用いることによって 2 章で述べた三種のバグすべてを検出できる。

### 3.4 専門知識モジュールの構成

専門知識モジュールは、教材である専門知識を用いて、学生への問題や解説の生成、学生の答えの評価、および、学生からの質問に対する解答を行う。図 5 は専門知識モジュールの構成を表し、太線で囲まれた部分が教材ごとに入れ替えられる知識である。

まず、学生の答えの評価は、問題を Prolog のゴールとして専門知識で実行し、得られた結果に基づいて行われる。

学生の質問のうち、「AgCl は沈殿しますか」等の yes/no 型の質問は、質問を専門知識で実行し、結果に従って解答される。「沈殿とはどういうことですか」等の what 型の質問に対しては、それに対応する解答

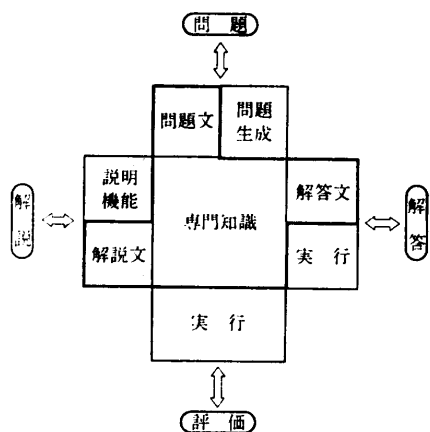


図 5 専門知識モジュール  
Fig. 5 Illustration of expertise module.

文を表示する。

学生への解説機能は、知的 CAI システムを教材に関するエキスパートシステムとみなした場合、専門知識の内容を説明するエキスパートシステムの説明機能となる。本システムの専門知識は、実際に教師が行う方法での問題解決、推論手法が記述されている透明な (articulate)<sup>2)</sup> システムなので、その問題解決過程をそのまま説明する機能であれば、学生にとって理解しやすい解説となる。また、この説明機能の他に、専門知識の各節に解説文を与えておき、これを表示する機能を用意している。実現の際には、各節の頭部の一つの引数にある知識番号をインデックスとして、解説文をファイルから検索している。先の解答文も同様に実現されている。

学生への問題は二種類に分けられる。一つは解答文や解説文と同様に教材作製者の用意した問題で、そのまま出題される。他の一つはシステムの持つ問題生成機能によって作られる問題である。この問題には、正しい問題 (例えば、答えが yes である問題) と、誤りを含んだ問題 (例えば、答えが no である問題) の二種がある。前者は、Prolog の検索機能により、引数をすべて変数化したゴールを専門知識で実行することによって生成される。後者は、専門知識モジュール内に問題生成のための知識を教材作製者に用意させ、それを用いて前者と同様に生成する。この知識は専門知識にバグを加えたものとなる。

### 3.5 個人指導モジュールの構成

個人指導モジュールは、教材作製者によって与えられた教育戦略知識に従って、システム全体を制御する。ここでは、実際の動作に従って、個人指導モジュール

の機能を説明する。まず、システムは学生に順々に解説や問題を与え、その応答から MIS によって学生の理解の状態をモデル化する。このとき、個人指導モジュールは、教育戦略知識に従い、学生への解説文や問題文の提示、あるいは、問題生成およびその出題を専門知識モジュールに指示する。出題に際しては、3.2.2 項で述べたように、問題を学生モデルと専門知識でそれぞれ実行し、異なる答えが得られるものを優先する。これは、モデル生成の効率化と、学生自身に自己の誤りに気付かせるという効果をねらったものである。専門知識モジュールはこの指示に従い、解説や問題を学生に与える。次に、学生のモデル化が進み、一定期間学生モデルに変化がない場合、モデル生成終了とみなし、PDS を用いてその時点での学生モデルと専門知識の比較を行い、学生のバグを検出する。検出されたバグは個人指導モジュールに送られ、個人指導モジュールは、そのバグを学生に気付かせ、直していくための問題や解説を専門知識モジュールに指示する。現在のシステムでは、検出されたバグが唯一の誤りの原因となるような問題を次々に与え、再教育を行っている。再教育によって学生がバグに気付いた場合には、モデルの再構築を行う。現在のシステムにおける教育戦略知識は、上記のようにモデル生成時の解説文・問題文の指定並びに問題生成機能への指示のみで、再教育のための教育戦略の取り扱い等の点で、十分な汎用化がなされていない。この教育戦略の汎用化は今後の研究課題の一つである。

### 3.6 対話インタフェース

対話インタフェースは、Prolog の内部表現と学生向きの表現形式との変換を行う。自然言語によるインタフェースは、最も期待され、研究が積み重ねられているが<sup>2), 8), 9)</sup>、本システムでは、現在は高度な処理は行っておらず、『Prolog 述語形式↔自然言語文』変換を行っているのみである。例えば、『chinden (Ag, Cl) ↔ AgCl は沈殿する』、『chinden (Ag, Cl) ↔ AgCl は難溶性である』、『chinden (Ag, Cl) ↔ AgCl は水に溶けにくい』等である。これらの変換のための辞書は、Prolog で書かれた知識として、対話インタフェースにおかれ、専用システムごとに入れ替え可能である。

## 4. 化学反応教育システム

本汎用知的 CAI システムに、化学反応 (酸と塩の反応) に関する教材知識を加え、化学反応教育システムを開発した<sup>10)</sup>。専門知識は図 6 に示すように八つの

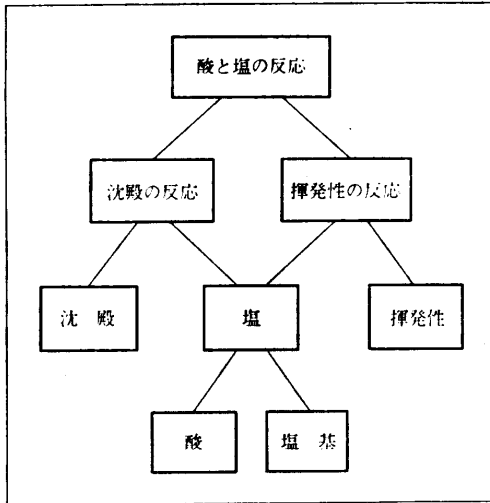


図 6 専門知識の階層

Fig. 6 Diagram of knowledge hierarchy.

これから一般的な酸と塩の反応について考えてみましょう。次の反応式が正しいときは yes、誤っているときは no と答えなさい。

①  $K(2)CO(3) + H(2)SO(4) \rightarrow K(2)SO(4) + H(2)CO(3)$   
 : yes  
 上の反応式をどのように判断しましたか?  
 理由は次の例のように文節単位に単文で答えて下さい。  
 例 HClは酸である。  
 AgClは沈殿する。  
 \*\*\*  
 (最後に \*\*\* を入力して下さい。)  
 : K(2)SO(4)は沈殿しない。  
 : H(2)CO(3)は揮発性である。  
 : \*\*\*  
 ②  $Na(2)SO(4) + 2HCl \rightarrow 2NaCl + H(2)SO(4)$   
 : no  
 ③  $Pb(NO(3))(2) + 2HCl \rightarrow PbCl(2) + 2HNO(3)$   
 : yes  
 ④  $BaCO(3) + 2HCl \rightarrow BaCl(2) + H(2)CO(3)$   
 : no  
 ⑤  $Ag(2)SO(4) + 2HCl \rightarrow 2AgCl + H(2)SO(4)$   
 : yes  
 ⑥  $2AgCl + H(2)SO(4) \rightarrow Ag(2)SO(4) + 2HCl$   
 : yes  
 ⑦  $PbCl(2) + H(2)SO(4) \rightarrow PbSO(4) + 2HCl$   
 : yes  
 ⑧  $CuCl(2) + H(2)SO(4) \rightarrow CuSO(4) + 2HCl$   
 : \*\*  
 : no  
 上の反応式をどのように判断しましたか?  
 理由は次の例のように文節単位に単文で答えて下さい。  
 例 HClは酸である。  
 AgClは沈殿する。  
 \*\*\*  
 (最後に \*\*\* を入力して下さい。)  
 : CuCl(2)は沈殿する。  
 : 逆反応が進む。  
 : \*\*\*  
 ⑨  $Ag(2)CO(3) + H(2)SO(4) \rightarrow Ag(2)SO(4) + H(2)CO(3)$   
 : no

図 7 教育例

Fig. 7 Example of a training trace.

世界から成る。最上位の酸と塩の反応の世界は、3.1.2 項で述べたように、下位の二つの世界（沈殿の反応、揮発性の反応）にある基本的な知識の適用に関するメタ知識である。また、下位の五つの世界は、それぞれ上位の世界で用いられる知識で、教材の階層性に基づくものである。図7は酸と塩の反応の世界にあるメタ知識の教育を行っている例で、教育している専門知識

```
rule(i_model,rule.1,X,Y,Z):-
    rule1(i_model,rule1,...X,Y,Z),
rule(i_model,rule.2,X,Y,Z):-
    not_rule1(i_model,rule1,...X,Y,Z),
    rule2(i_model,rule2,...X,Y,Z),
not_rule1(i_model,rule.3,X,Y,Z):-
    not(rule1(i_model,rule1,...X,Y,Z)),

rule1(i_model,rule1.1,X,Y,Z):-
    en(i_model,en,...X,Y),
    chinden(i_model,chinden,...X,Z),

rule2(i_model,rule2.1,X,Y,Z):-
    en(i_model,en,...X,Y),
    kihatusei(i_model,kihatusai,...Y),
    hukihatusei(i_model,kihatusai,...Z),
```

図 8 専門知識

Fig. 8 Contents of expertise.

```
rule(s_model,rule.1,X,Y,Z):-
    rule1(s_model,rule1,...X,Y,Z),
rule(s_model,rule.2,X,Y,Z):-
    rule2(s_model,rule2,...X,Y,Z),

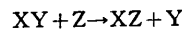
rule1(s_model,rule1.1,X,Y,Z):-
    en(s_model,en,...X,Y),
    chinden(s_model,chinden,...X,Z),

rule2(s_model,rule2.1,X,Y,Z):-
    en(s_model,en,...X,Y),
    kihatusai(s_model,kihatusai,...Y),
    hukihatusei(s_model,kihatusai,...Z),
```

図 9 学生モデル

Fig. 9 Constructed student model.

の一部を図8に示す。問題①から問題⑥までがモデル生成のための問題で、モデル生成終了時の学生モデルの一部を図9に示す。専門知識・学生モデル中の各述語の第一～第三引数は、それぞれ、専門知識と学生モデルの区別、世界名、解説文等へのインデックスである。述語 rule, rule1, rule2 は、上位三つの世界での反応を表す知識で、第四～第六引数 X, Y, Z は、反応式



塩酸 塩酸

に対応する。この学生は、3.1.2 項で述べた例のように、逆反応が起きる場合を考慮していないので、問題⑥に対して yes と誤答している。システムは、図9のように逆反応を考慮しない学生モデルを生成し、そのバグに基づいて、逆反応が起きる例題を次々に与え（問題⑦から問題⑨）、再教育を行っている。自身の誤りに気付いた学生は、問題⑧に対し考えの変更を指示し（入力「\*\*」）、その後で、正答と変更した考えを入力している。

### 5. むすび

従来の知的 CAI システムにおけるモデル生成法の欠点を考察し、その欠点を補う新しい汎用知的 CAI システムの枠組と、その応用事例として化学反応教育システムについて述べた。本汎用知的 CAI システム

は、スーパーミニコン MV/8000 II 上に、MV-Prolog を用いて実現している。実現言語に Prolog を用いたことにより、モデル表現言語と合わせ、Prolog で統一されたシステムとなっている。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に基づくモデル表現
- 2) 帰納推論によるモデル生成
- 3) 教材に依存しない汎用性

4章の教育例にみられるように、学生のバグを教材に依存しない形で、正確にモデル化するという点での本システムの有効性は示された。現在、本システムの汎用性を確認するために、化学反応教育システムと並行して、Prolog プログラミング教育システムの開発を進めている<sup>11)</sup>。今後、本システムは知的 CAI システム開発支援ツールとして完成させていく予定である。この点での今後の課題としては、本文中に述べた教育戦略の汎用化の他に、対話インタフェースの充実と教材作製者用のユーティリティの作製が挙げられる。

謝辞 日頃、熱心なご討論、有益なご助言をいただき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に深く感謝します。

#### 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) Angluin, D. and Smith, C.H.: Inductive In-

ference: Theory and Methods, *ACM Comput. Surv.*, Vol. 15, No. 3, pp. 237-269 (1983).

- 4) Clocksin, W.F. and Mellish, C.S.: *Programming in PROLOG*, Springer-Verlag, New York (1981).
- 5) 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).
- 6) Brown, J.S. and Burton, R.R.: Diagnostic Models for Procedural Bugs in Basic Mathematical Skills, *Cognitive Science*, Vol. 2, No. 2, pp. 155-192 (1978).
- 7) Shapiro, E.Y.: *Algorithmic Program Debugging*, MIT Press, London (1982).
- 8) Sleeman, D. and Hendley, R.J.: ACE: A System Which Analyses Complex Explanations, in Sleeman, D. et al. (ed.), *Intelligent Tutoring Systems*, pp. 99-118, Academic Press, London (1982).
- 9) Clancey, W.J.: Tutoring Rules for Guiding a Case Method Dialogue, in Sleeman, D. et al. (ed.), *Intelligent Tutoring Systems*, pp. 201-225, Academic Press, London (1982).
- 10) 喜納他: 帰納推論に基づく知的 CAI システム—化反応学を教育する場合—, 電子通信学会オートマトンと言語研究会資料, AL84-44 (1984).
- 11) 願化他: 帰納推論に基づく知的 CAI システム—PROLOG 教育の場合—, 情報処理学会知識工学と人工知能研究会資料, 38-2 (1985).
- 12) 河合他: 帰納推論を用いた ICAI システム—汎用システムアーキテクチャの設計—, Proc. of the Logic Programming Conference, 15-4 (1984).

(昭和 60 年 3 月 18 日受付)

(昭和 60 年 5 月 9 日採録)