報　告

パネル討論会

知的 CAI の実現—開発経験者のノウハウを中心に理論から実現まで—「教育における知的方法」シンポジウム† 報告

パネリスト
竹内　章[1]，　松田　昇[2]，　山本　秀樹[3]
渡辺　成良[4]
司会　大槻　説乎[5]

大槻(司会)　それでは，時間がまいりましたので，パネルセッションを始めたいと思います．

最初に，パネルセッションの目的などを話しまして，各パネラにご自分がおつくりになったシステムのセールスポイントだとか，難しかったところとかを中心に，残された問題も含めてご説明いただきます．

一とおり発表が終わりましたら，フロアの方から質問を受けたいと思います．お話の内容と無関係なことでも結構です．

まず，このパネルディスカッションを計画した意図ですが，最近日本で知的 CAI に興味をもっていらっしゃる方が増えてまいりました．新しい興味のある理論もたくさん出てきて非常におもしろくなりましたが，理論は発表を伺えばよく分かって，ディスカッションできますけれども，作ったシステムはディスカッションの対象にはなりにくいものですし，理論的な面とは別に聞きたいところもあるのではないかというのが，パネルを企画いたしました目的の一つです．

1990年に知的 CAI の国際会議が東京で開かれます．知的 CAI とは限りません．新しい教育環境をテーマとする国際会議が開かれますので，いろんな興味深い理論をおもちの方も論文を発表されることはもちろん，ぜひそれを実現して展示していただきたいと願って今回のパネルセッションを計画いたしました．

私，実はこの間，教育に関する環太平洋国際会議に出まして，そのキーノートアドレスで，アメリカのアンダーソンさんのグループの人たちが実現した ITS (Intelligent Tutoring Systems)* を，子供が使っているビデオを見せていただきました．それは予想したとおり，実用を目的としたシステムで，いろいろ問題もあるのですが，大変よく分かったわけです．

そのときに，興味のある話が実は一つございました．このシステムを使って学習した子供の理解度をグラフにしたものを見たのですが，4つのグループに分けて，数値が示されていまして，一番低い数値のものが，伝統的教室集団授業といいますか，要するに日本で先生が子供に教えているのと同じ一斉授業でございます．その上が，伝統的 CAI でして，その上が，アンダーソンのシステム，一番上に人間による個人チュータというのがありました．

たぶんアンダーソン先生の実験の結果だと思うのですが，数値は一番下が 50 余りですね，つまり 50% 余りの理解を得られた．伝統的 CAI でやったら 70% そこそこ．実はきちんとした数値が示されていたのを，私が忘れているだけで申し訳ありません．3段目の ITS で八十数％となかなかいい数字が出ておりまして，一番上の人間が1対1で一生懸命教えるところで九十何％出ているんですね．人間が一生懸命1対1で教えても，なお分からない部分が残るという話でございます．

それからもう一つ，去年，アーティフィシャル・インテリジェンス・エンサイクロペディアがアメリカで出版されました．知的 CAI の部分を読みまして，印象に残ったことが二つあります．

一つは，どういうものを対象に知的 CAI をつくる

† 日時　昭和63年11月10日 (木) 16：30～18：30
場所　機械振興会館大ホール (地下2階)
1) 九大，2) 金沢工大，3) 神電気，4) 群馬大，5) 九工大

* ITS は，知的 CAI と同じ意味に用いられる．現在市販されている CAI との混同を避ける意味で知的 CAI のことを ITS，従来型の CAI を伝統的 CAI と呼ぶことが多い．本文中に出る専門用語は情報処理 29 巻 11 号「知的 CAI 特集」を参照されたい．

かということです．知的 CAI は何でも対象にできると書いてあるんですね．つまり対象を選ばないと．やりやすい，やりにくいはもちろんあるんでしょうが．日本で作成されているのは，対象がかなり限られているのが現状ですので，大変私，印象に残っております．

もう一つは，実用化の見とおしという項がありまして，カースレーさんのご見解だろうと思いますけども，だいたい10年というふうにみておりますね．今でももちろんアンダーソン流のものは実用に近いわけですが，そうじゃなくて，今理論として提起されているものが実用的なシステムとして普及するのに，10年かかるであろうと．その最大のネックは，ITS をやってる人と伝統的 CAI をやっている人の間の断絶にあると書かれています．ITS をやっている人は実用化に対してあまり熱心でなく，伝統的 CAI をやってる人は，ITS についてはまったく興味がない，そこが最大のネックで，そこを埋めることによってもっと早くなるというふうに書いてありました．

これ以外にいろんなキーポイントがあると思いますが，それはお話の中でご披露していただくことにいたします．あくまでも本当にお作りになった人がしゃべるということでございます．

では，まず竹内先生お願いいたします．

1. ITS のフレームワーク

竹内 九州大学の竹内です．

まず，システムの名称，使用したハードウェアからということでしたので，そこから話をはじめます．

システムの名前は Book で，約10年前から研究を続けています．Book は，ITS のフレームワーク作りを目標にしていますので，教授対象領域は特に限定していません．教授内容に依存した知識だけを入れかえれば，いろいろな教科の知的 CAI 教材が利用できるようになる枠組みに関する研究を中心にしてきました．教材の例としては，引き算，Fortran の入門，清書システムの使い方などがあります．

使用したハードウェアは，一つは超大型のメインフレームコンピュータ，もう一つはごく普通の 16 ビットパソコンです．これは両極端なんですが，それぞれで独立に動くシステムを作るだけでなく，同じプログラムを両方で動かそうと試みたりもしています．言語は Prolog を使っています．

ITS を構成する要素には，一般に，教材知識，教授知識，学習者モデル，対話インタフェースがあるといわれていますが，Book もこれらの要素をもっています．それぞれについて特徴を簡単に紹介します．

まず，知識の表現方法としては，私たちのところで多重階層モデル (MHM) という表現を考えて，これで教える中身に関する知識を書いています．多重階層モデルでは，教える中身，いわゆる教材知識を幾つかの世界に分けて記述します．それぞれの世界には，インタフェースに使う辞書のための世界が一対一に対応しており，さらにそれらとは別に文法規則の世界があります．教材知識である概念域と呼ぶ部分と，辞書と文法の世界である言語域と呼ぶ部分は独立にできてますので，概念域を1個つくっておけば，言語域を入れ替えて，日本語で対話したり，英語で対話したりということができるような枠組みです．

対話インタフェースの入力処理ではちゃんと構文解析をしているんですが，出力のときにはテンプレートを使って特定の知識を名詞句に直すとか，文章に直すとか，あるいは個々の知識を図形に変換するための手続的な規則を書いておいて，図形を合成するとかして出力しています．

教材知識を幾つかの世界に分割しているんですが，MHM では分割した世界の間に，世界の中に含まれている知識の関係に応じて，幾つかの知識の継承関係を定義するようになっています．その一つに，ストラテジグラフという教える順番に関する関係があります．これで教える内容に関する知識の構造を明示しておきます．そしてこれを利用して学生を指導するときにどの知識を使って指導すべきか，次にどこの知識を教えるべきか，ということを選択する枠組みになっています．

引き算の例ですと，1桁の引き算をまず教えてから，次に繰り下がりの練習があって，そして2桁の引き算に進むとなっています．1桁の引き算などもさらに細かく幾つかの世界に分かれてます．

こういうふうに知識を細かく幾つもの世界に分けておいて，世界間の関係を明示しておき，推論に使用する世界を制御することの利点というのは，学生を指導するときに非常に生きてきます．

学生を指導するときというのは，たとえば，教育目標である，今教えたい知識を使うような問題を学生に与えないといけないわけです．問題をあらかじめ用意

しておくのではなくて，システムが自動的に生成しようとしますと，教育目標である知識を使って，その知識を使うような問題を生成するとか，あるいは過去にシステムが多くの学生を教えて，どうも共通して間違いやすい知識があるということをシステムが学んでいきますと，その知識を問題解決に使うような問題を生成するとかの必要があります．

そういうときは，特定の知識を指定して，その知識を使わないと解けない問題を生成するんですが，指定した知識以外はどれを使ってもいいわけです．ただ，教えてない知識を使ってはいけないわけですから，何を教えてあるかという情報が必要になります．教材知識の世界は，難易度に応じて分けてあります．ですから，たとえば減加法で十幾つ引く１桁の数の引算を教える場合に，答えが５以下になる知識だけを使って質問を生成すれば，やさしい問題ができる．６から８の数に３以下の数を加えるという，より難しい知識も使って質問を生成すれば，より難しい質問が生成される．こういうぐあいに学生の状態に応じて，生成する質問の難易度とか，使う知識を自由に制御できるという枠組みが MHM で得られています．

次に学習者モデルに関してですが，学習者の誤り原因を同定する方法として，私たちは摂動法という方法を提案しております．これは，正しい教材知識に対して教育的にありそうな，実際学生がやりそうな間違い現象に対応した摂動をかけて誤りの仮説をつくり，その仮説で学生の誤った答えを導出できるかどうかテストして，誤り原因を同定するというやり方です(図-1)．

誤り原因を同定すると個人の学習者モデルができます．それを集めてバギーカタログをつくり，平均的な学習者モデルをつくって，ここにシステムの教授経験を蓄えています．バギーカタログは，摂動法によってシステムが自動的につくることもしますが，こういう間違いがよくあるんだということが分かっているならば，必要に応じて経験のある先生があらかじめ初期値として与えておいてもいいわけです．

指導方法としては，適応指導という方法を提案しています．これは，学生の間違い原因を摂動法なり，バギーカタログで同定したあと，それを指導するときに，よく分かっている学生に対しては，学生が自分で間違いに気がつくように，学生の誤りを直接ただすんじゃなくて，誤りを間接的に指摘し，あまり分かってない学生に対しては直接誤りを指導するという方法です．これは，学生の理解のレベルに応じて，指導対象となる知識の抽象化のレベルを変更するわけですが，そのレベルを制御する方法として，適応指導を提案しています．

次に，システムをつくる段階でうまくいったことですが，一つだけ，Prolog のご利益といいますか，Prolog を使っている方ならば感じておられることを，あげておきたいと思います．

CAI システムでは，たとえば，誤り原因を同定するときに，正しい知識を書き替えて間違った知識をつくったり，質問を生成したりする場合に，教材知識を本来の問題解決とは異なった推論方式で使います．こういう場合に，Prolog は構文が簡単で意味が明確ですし，オブジェクトレベルとメタレベルのプログラムが容易に両立しますので，都合がよいわけです．

それから自然言語の入力インタフェースが比較的簡単につくれる．私たちは，現在，英語は DCG で，日本語は BUP で書いてますけども，これは Prolog と相性がいいですから，システムの中に組み込むことが容易であるという利点があります．

最後に，今後に残された問題ですが，学生の間違い

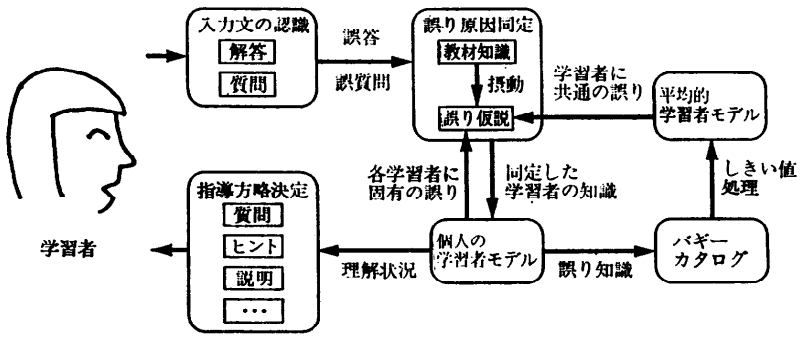


図-1 摂動法による学習者モデルの推定と指導

がどういう原因で，どこで生じたかということを同定したあと，どう指導するかという問題があります．指導方法として私たちのシステムでは，誤りの場所を示すとか，内容を示すとか，誤った知識に関する性質について質問をしたり，反論したり，はじめに与えた問題よりも単純化した問題を生成して与えたり，あるいは単純化して学生がすぐ間違いに気がつくであろう反例をシステムが生成して与えたり，単純化した例をそのまま説明してみたり，というような方法を用意して，実現しています．ところが，教えている内容，それから知識のどこをどういう原因で間違ったかによって，どの指導方法を使うべきかということが，実は今のところ明確になっておりません．それをどう決めるかが，残された大きな課題です．

大槻 ありがとうございました．引き続きまして，金沢工業大学の松田先生にお願いしたいと思います．

2. 幾何を教える ITS

松田 岡本先生と共同研究をさせていただいた松田です．

岡本先生の研究室では，一番初めにオーサリングシステム IROSA を開発しました．それから，初等微分演算を扱った NEUTON というシステムを開発しました．これは，学習者モデルに MOD モデルを採用しておりました．

その次が，私が岡本先生と一緒に開発したシステムで，初等幾何を扱った GEOMEX です．本日はこのシステムを中心にお話をします．学習者モデルは，オーバレイモデルと差異モデルを併用しております．

現在は，初等積分演算を扱った ITS が開発されています．これは学習者モデルにオーバレイモデルとバグモデルを使っているようです．使用言語は，Prolog および Lisp です．

本日お話します GEOMEX は Lisp を使って構築しました．GEOMEX は中学校２年生程度で学習する三角形の合同の学習世界における ITS です．

GEOMEX では，学習者が自由な形式で証明を行い，その入力された証明をシステムが認識します．そこで学習者がどういった証明計画に基づいて証明しているかを推論します．その結果に基づいて，適切な教授戦略を決定して，それに基づいた助言を行う．こういった枠組みの ITS を目標に研究を開始しました．

そこで，具体的にはこのような構造を考えてみました (図-2)．計画認識モジュールで学習者モデルを生成します．その結果に基づきまして，教授戦略を決定します．教授戦略は，今回のシステムでは，対話手続き知識ベースという知識ベースに基づいて，これはIF-THEN 形式のプロダクションルールで書いてありますが，それによって学習者モデルを見ながら決定されます．本日は，実際にどういった方法で計画認識をするか，ということを中心にお話をしたいと思います．

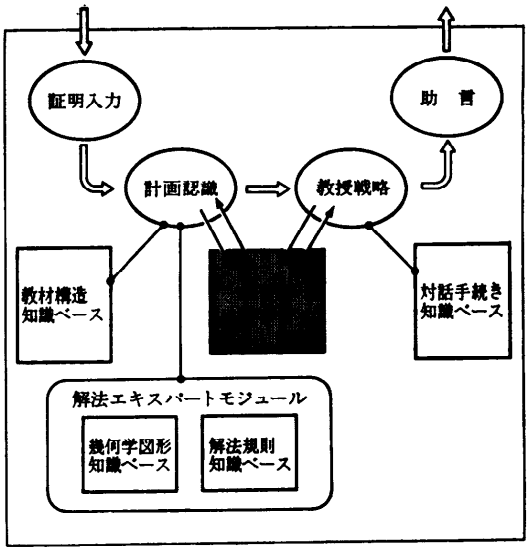


図-2 GEOMEX のシステム構成

われわれは初め，教材構造知識ベースというものを用意して，これを用いてオーバレイ的な考え方を使い，計画認識をしようと考えました．

教材構造知識ベースと言いますのは，たとえばある特定の問題を証明するために，幾つかのサブゴールがあるとします．一つは三角形の合同を使う．もう一つは辺が等しいことを利用するといった事柄です．そのサブゴールを証明するためには，さらに幾つかのサブゴールがあるという，特定の問題を解くための理論的に可能な証明方法を木で記述しているものです．

こういった木を個々の課題ごとに用意するわけです．この木の中で学習者がどういった証明をしているかを，システムがオーバレイ的にみてその結果，学習者の証明結果を認識するといった方法です．

たとえば，学習者が三角形の合同ということを利用して，「辺が二つ等しいから角が等しい」といったことを想定している場面の対話例を考えます．

システムは教材構造知識ベースをみますと，確かに三角形の合同を利用すれば，それらの前提から AB=

AC が言えると分かるわけです．さらに合同条件として，二辺夾角という公理を使えば確かに角が等しいと言えることが分かるわけです．そこでシステムは，「あなたの考えは分かりました．しかし，もう少し詳しく証明するべきです．この三つから次のどれが言えますか？」と助言します．

そこで，学習者が自分の考えている三角形を正しく入れたとします．そうしますと，次にシステムは学習者に，合同条件を質問します．『三角形の合同が証明できそうですね．何という合同条件を使いましたか？』これは実際には番号で入れさせます．そのあと続いて三角形の合同が言えて，あなたの考えているAB＝AC が言える，という対話ができる仕組みです．

しかし，こういった方法だけを用いて計画認識をしていますと，この教材構造知識ベースは正しい解法プロセスしか表現していませんので，学習者の間違った証明計画を認識することが困難なわけです．

そこで，その次にわれわれは，解法エキスパートを利用する計画認識の方法を考えました．解法エキスパートは汎用のプロダクションシステムですが，そこに幾何の世界で使う解法規則と，幾何学図形を表現するための知識ベースをもたせました．こういった専門家モジュールを使って，学習者の教材構造知識ベースには存在しない誤った証明方法を認識することを試みました．ここで差異モデル的な考え方を取り入れたわけです．

その計画認識の方法を次に説明します（図-3）．学習者は，普通幾何の証明をするときは，幾つかの前提からある事柄が言えるという形式で証明を進めていきます．先ほどの場合ですと，辺が等しいという二つの前提と，角が等しいという一つの前提，つまり三つの前提から，辺が等しいということを主張していたわけです．

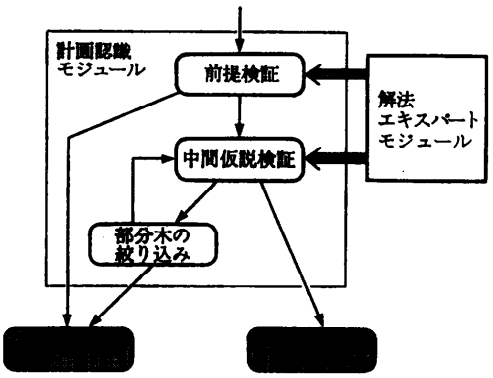


図-3 計画認識モジュールの構成

こういったステートメントが入ったときに，計画認識モジュールが起動します．

まず，学習者の主張してる前提を検証します．教材構造知識ベースにない場合には，解法エキスパートを使って，前提が言えるかどうかを調べます．つまり，その前提をゴールとして解法エキスパートに与えて，証明ができれば正しいわけです．こうして，すべての前提が正しいことが分かった場合には，それから，学習者の主張している中間仮説が言えるかどうかを，再び解法エキスパートを使って検証します．

もしそれが実際に証明できれば，システムの推論した証明プロセスを学習者も考えているという仮説を立てます．もしそれが失敗した場合には，再び教材構造知識ベースを参照して，部分木の絞り込みを行い，もう一度中間仮説の検証を行います．

学習者の主張が誤っている場合には，学習者の主張している前提が教材構造知識ベースの中でばらばらに存在しています．ばらばらのノードから学習者がある中間仮説を考えているという状況になっています．部分木の絞り込みとは，学習者の主張してる前提をもっとも多く含むような部分木を同定することを言います．そこでシステムは，部分木から学習者の主張している結論が出るかどうか解法エキスパートを使って検証します．もしそれが成功した場合には，学習者は誤った証明計画を立てている可能性があるわけです．

今言いましたのが，部分木の絞り込みですが，それでも学習者の主張する証明プロセスが見つからない場合は，誤った証明だとして，その誤りを説明します．

たとえば，三つの前提より角が等しいと学生が考えているとします．こういった証明方法が教材構造知識ベースになかったとします．つまり誤っているわけです．システムはこの三つの前提から，角が等しいと言えるかどうかを，解法エキスパートを使って推論します．その結果，解法エキスパートは，三つの前提から三角形の合同を考え，それを利用して，さらに角が等しいと考えれば，証明が成り立つという結論を出し，学習者も同じ計画を用いているという仮説を立てます．

これはあくまでも仮説ですから，学習者に，本当にそうかどうかを質問するわけです．ここで実際に学習者が，イエスと答えたことにします．しかしその証明は正しくないので，次にこの証明がどうして誤っているかを説明することになります．この例ですと循環論

法になってしまうわけです．つまり前提の一つはこの問題の仮定でなかったわけですからこの前提を証明する必要があります．しかし，証明できません．なぜならば，証明するためには，今中間段階で使った三角形の合同を利用する必要があるので，循環論法になるわけです．システムは，教材構造知識ベースを参照して，それが循環するかどうかを調べます．

このようにして学習者の証明計画を認識しながら，対話を繰り返し，学習が進行していきます．

最後に，インタフェースの話ですが，本システムでは，証明で使うキーワードをファンクションキーに用意しておいて，そのキーワードの組み合わせで証明を記述することになっています．現在は 20 ぐらいのキーワードがあります．以上です．

大槻 どうもありがとうございました．

それでは，続いて沖電気の山本さんにお願いしたいと思います．山本さんは，場面を限った英会話の知的 CAI を作成されました．会話文のほかに絵も出るし，アウトプットだけですけど声も出るというものです．

3. 英会話を教える ITS

山本 ただいま紹介にあずかりました山本です．私は沖電気の総合システム研究所で知的 CAI システムの研究を行っています．

本日は英会話教育用の知的 CAI システムについて話をさせていただきます．

題材に英会話を取り上げたところが，他の方とはちょっと違うところですので，少しそのバックグラウンドについて説明します．

現在英会話学習の教材としては，テープやビデオなどがあります．ここにいらっしゃる皆さんもこういった教材で勉強された方がたくさんいらっしゃるのではないかと思いますが，こういった教材というのは，基本的にパターン化された会話の記憶が中心となっています．しかし，実際の会話は，教科書のようにパターン化されていませんので，覚えたとおり話をしても，会話相手の外国人は教科書どおり言ってこないのではないかという不安感がなくなりません．勉強しても自信がつかないので，使わなくなってしまって上手にならない，ということが多いのではないかと思います．

そこで，われわれは AI の技術を使って会話をシミュレートすることによって，人間と会話をしているような状況に学習者を置いて教育するシステムを作ろうと考えました．英会話をシミュレートする ICAI システムには，以下の機能が要求されます．

まず，システムが会話の流れを理解して，学習者の入力に誤りがあっても，会話を中断せずに続けることが必要です．

次に，学習者が話題を変化させても，システムはそれに対して追随しなければなりません．

3番目に，学習者のレベルに合わせた会話を行ったり，間違いなどに対して教育的なアドバイスを行ったりする教育的機能が必要です．

最後に，即時の応答が必要です．1回の入力に対して1分たって応答が返ってくるようでは，会話をシミュレートしているとは言えなくなってしまいます．

このようなシステムでは教育的な機能や応答の即時性も，根本的に会話のシミュレーションの方式や性能に関係してきます．

そこで，どうやって会話のシミュレーションを行うかについて簡単にお話します．

まず考えたのが，文の表層情報だけから会話を行うという方法です．イライザというシステムで使っている方法ですが，その方法を使って会話のシミュレーションを実現しますと，システムとの会話は非常に単調になってしまって，会話の訓練に十分ではないと判断しました．

そこで，われわれのシステムでは，目標志向の会話のモデルを採用しました．会話には目標があって，それを達成するために会話者が情報を交換するというのが，目標志向の会話でして，双方が主導権をもつ会話が可能です．こういった特徴がありますので，これで会話の訓練を十分に行うことができるだろうと考えました．

次に，学習者モデルの話をします．英会話には，単語，熟語，構文，文法，会話の状況といった知識が必要です．このシステムでは，学習者の入力文の誤りから学習者の理解状態を把握するためにバギーモデルを採用しました．システムは誤りを同定すると，学習者データを変更します．そのデータは出力文のレベルを変えるのに使っています．

ここまでが内側の話でして，先ほど大槻先生がおっしゃった外側の話をさせていただきます．

システムは，パソコン，LISP 専用マシン ELIS，音声合成装置からなります．学習者はパソコンに対して入力を行います．ELIS の上では，会話のシミュレー

ション機能，教育的な機能を実現しています．会話の応答は，パソコンの画面に表示されつつ，音声装置から音声になって出力されます．

ELIS の上では，入力文理解部が構文解析，意味解析を行って，入力文を内部表現に変換します．その内部表現は状況メモリに入り，それを見て会話制御部が次の発話を決め，出力文生成部が英文を生成するという流れで一つの会話が成立します．

入力文の誤りは学習者データに反映させ，それによって出力文のレベルを変えています．

次に，システムの動作をご紹介します．ホテルのフロントの状況を示す絵がパソコンの画面に表示されます．フロントマンをシステムが，そして旅行者を学習者が演じて会話をするところを説明しましょう．

まず，システムが「May I help you?」という質問をします．この状況で学習者が，たとえば「I want to staying at this hotell」とキーボードから入力します．そうすると音声合成装置がそのように発音します．この入力は，ご覧になると分かりますように（図-4），staying のところと，hotell のところが誤っています．

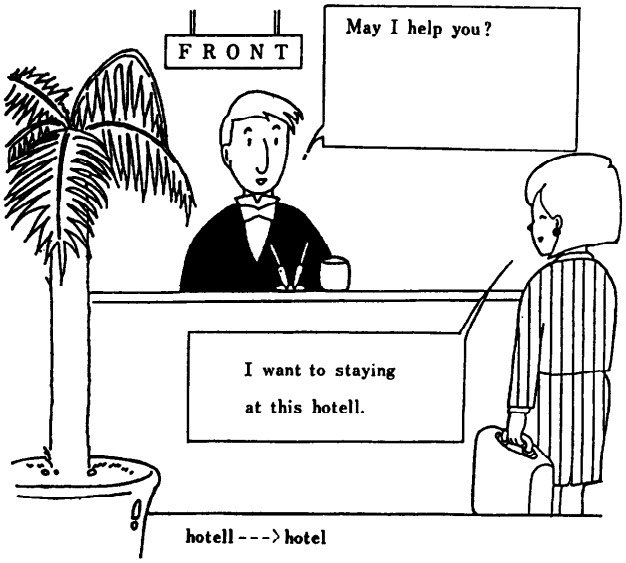
システムは，入力文理解部でこの誤りを発見して，hotel の綴りが間違っていることと，to の直後の動詞は原形でないといけないことを学習者に指摘します．

引き続いて意味を解析して「泊まりたい」という意味を判断し，話題を予約に関する事柄に変えて「Do you have a reservation?」という発話をします．

もう一つの例を見ましょう．先ほどの話が進んだところの会話で，「Fill in this registration card, please.」とフロントマンが言ってきたところです．ここで学習者がたとえば，この “registration card” の意味が分からないとします．そのような場合「What is “registration card”」と入力しまして，ダブルクォーテーションに囲まれた言葉の意味を尋ねることができます．するとシステムは，「この語の意味は宿泊台帳です」という教育的メッセージを出力します（図-5）．

今後 の課題としては，より高速の会話理解方式の開発，文法辞書知識の整備拡充を考えています．

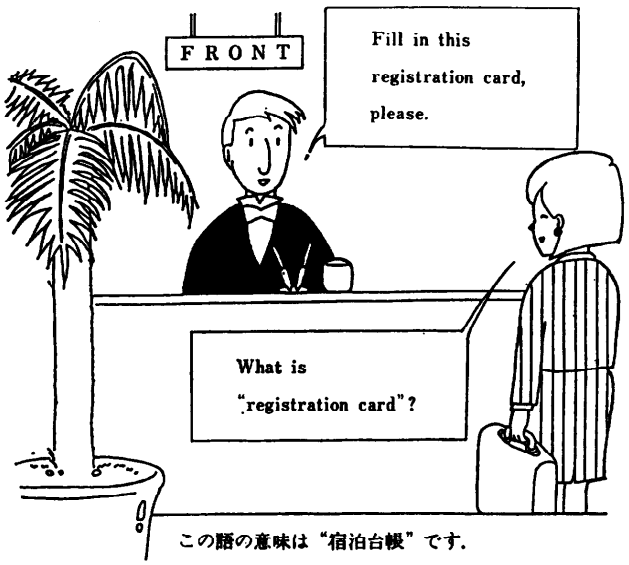
最後に，おもしろかった点ですが，「May I help you?」とシステムが話す画面をいろいろな人にお見せして「入力してください」と言いますと，ある程度の入力は予測してこちらのほうも最初から知識を用意していますが，まったく予想もしない入力が多くみられました．その中で多かったのが「Do you know my



hotell ---> hotel

to の直後の動詞は原形です．

図-4



この語の意味は “宿泊台帳” です．

図-5

name?」です（笑）．この種の入力が多くて最初は戸惑いしました．しかし皆さんシステムの知識に対してはかなり疑いをもって挑んでこられますが，システムと会話をすること自体には好意的な態度でした．この点が開発者としておもしろく感じられました．以上です．

大槻 どうもありがとうございました．

それでは最後に，群馬大学の渡辺先生にお願いしたいと思います．渡辺先生は先ほどご発表いただきましたので，システムの内容についてはもう皆さんご承知かと思いますが，長い時間かけてこのシステムをおつくりになられて，改良を重ねてこられた経験がおありですから，ぜひそこを伺ってみたいと思います．

4. 回路解析を教える ITS

渡辺 システムに名前をつけてくださいと言われてどうしようかと思ったんですけれども，アレン（ALLEN）としました．これは私の子供に，ミドルネームとしてアメリカの友人が付けてくれた名前なんですけど，ちょうどこの回路解析支援システムに合うので選びました．子供が大きくなって，このシステムで勉強したほうが私が教えるよりはよかったというふうになればいいなと思うんですけれど．ALLEN は，抵抗回路や交流回路の問題を学習者に提示して，枝電流解析，ループ電流解析，節点電位解析の三つの解析法を理解してるかどうかを診断します．このために一応複素数の演算もできる，代数方程式が解けるという，そういう機能はもたせないといけません．使用機器は，現在 CPU が MC 68000 で，2メガバイトメモリ，8インチのフレキシブルディスクを2台もっています．使用言語は Prolog です．

システムの開発過程を学会の発表順にたどってみます．最初は 1977 年，数式処理のためのハードウェア構成をどうしたらいいかという問題を学生といろいろ考えて，DEC 社 LSI-11 というマイクロプロセッサを手に入れました．この上で Lisp を走らせてみて，少しやれそうな感触を得ました．

そのころ，Simon Collins 監修の本を淵先生らが訳され，人工知能の基礎として出版された中で，SOPHIE という知的 CAI システムを知りました．私は回路演習をちょうど受け持っておりましたので，解答を添削するシステムをつくろうとそのとき考えました．

その理由に添削は実時間処理を必要としない点がありました．SOPHIE のような対話型 CAI は私たちのハードウェアでは実現できそうになかったからです．

さて，答案添削では記述された文章を解析しないといけませんから，学生に答案の記述に関していろいろ調べさせました．当時は，AI における自然言語処理や格文法の知識が私たちにはなかったので，教科書の中の単語を取り出して辞書をつくったり，文法規則を調べたりしたのが手探りの時代，1980 年ごろです．

ようやく 1982 年に，ハードウェアが好きな学生が来たものですから，Z 80 を5台スター結合した計算機を作りました．そして Lisp で数式処理プログラムを書き，並列処理の利点を調べました．たまたま Z80 で走る Micro-Prolog の存在を田中穂積先生に教えていただき，すぐに Prolog にかえました．1983 年です．

なぜ Lisp から Prolog に変えたかと申しますと，大学では学生が毎年代わっていきます．そうすると，ソフトウェアの保守で非常に困るんですね．というのは，新しい学生が先輩の作った Lisp プログラムを理解するのに，半年以上かかってしまうということがあります．それでは全然仕事になりません．Prolog は，そういう点の問題は少ないという話を伺ったものですから，Micro-Prolog を勉強して，Prolog に対してよい感触を得ました．そこで回路関係の知識を組み込む計画を実行に移しました．答案添削は 1986 年情報処理学会で発表したシステムで一応実現できました．

ここで答案に含まれた誤りが，その後の解答に波及して生起した誤りを推論する問題が残されました．そういう推論方法はなかなか難しそうだということで，対話型に使うシステムにこの時点で変更しました．それで，きょう発表させていただいたようなものになったわけです．

そういう経過をたどっておりますので，私どものシステムの特徴は答案添削にあります．学習者の記述した答案がどういう過程をとって解を導くかを調べる方法がどこまで構築できるかという問題を提起し，そういう学習者主導型システムを目指しました．答案は日本語文と数式で記述させますので，文解析とシミュレーションの機能をシステムにもたせる．それから，先ほど説明いたしました数式処理と対話の助言機能，そういう機能が不十分ですけれども，動くものになっているというのが現状でございます（図-6）．

これまでの研究課題は現実に使えるシステムを実現

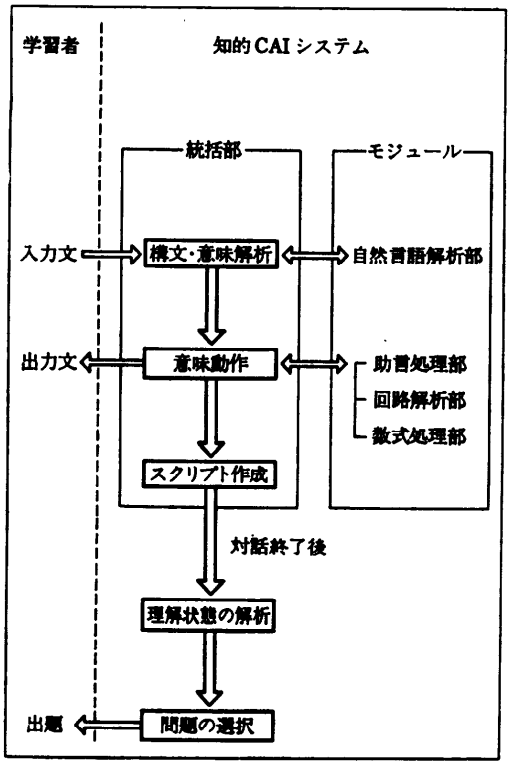


図-6 ALLEN 概念図

することでしたが，大阪大学の豊田先生や角所先生のグループのように，今後は理論的な裏付けのもとに，どこまでシステムを実現できるかというのにチャレンジしていくのが，大事なことではないかというのを実感しました．たとえば入力文解析の最初のバージョンは自己流のために，入力文の処理時間が9秒とか，11秒でしたが，これは構文解析を BUP に直しましたら，同じシステムで5秒程度に下がります．ワークステーションに変えると，1秒以内に終わってしまう．急がば回れで，理論を大切にすべきだと感じました．

さて，学習者がつくった式の正誤を診断するためにシステムはシミュレータにより実際に内部で式をつくります．シミュレータは学習者が記述する解き方に従って式をつくり出しますので，学習者がつくった式とその式とを比較するという機能も，数式処理にあります．

最後に，私どものシステムの問題としては問題をどうつくるか，解答過程を記述するスクリプトをどうするか，すなわち教科内容とスクリプトと問題のつくり方，この辺を解決する必要があります．

大槻 どうもありがとうございました．

4人の方のご報告を伺いますと，対象領域といい，知識表現といい，インタフェースといいまったく違っています．当たり前のことですが，自由になんでもつくれるというのは，意を強くするところだと思います．プログラムというのは本来そういうものですけれども，ITS も，やりたいように好きに設計して，好きにつくればいいことを実証できたんじゃないかと思います．

ここで皆さん方のご質問を受けたいと思います．

5. ディスカッション

質問者 1 論理マシン，いわゆるノイマン型のコンピュータで実際につくられているわけですが，その限界，あるいは教育においては教師の直感力や生徒の直感力というものが非常に重要な問題になっていますが，そういった直感をノイマン型のコンピュータで載せることができるかどうか，その辺に関して先生方に何か知見がありましたら，お教え願いたいんですが．

大槻 直感力はどの場面で必要になりますか．

質問者 1 教師が指導する上で，実際にノートに書いている状況を見たとき，こいつは分かっている，ここは間違っているといった教師の直感を実際に指導する上で使っている場合があるんですが，そういったことを ITS で実現できるかどうか，お聞きしたいんですが．

大槻 論理的に間違っているとか，合っているとかということではなく，直感的に合っている，間違っているというのが実現できるかという話ですか．

質問者 1 そうですね．

大槻 じゃどなたか．

渡辺 直感といっても，こういう答えはいいとか，悪いとかを理論的に評価したことがあっても意識していないだけの直感か，それとも，見たこともないようなことで，答えが合っているから直感として正しいという考え方のどちらでしょうか．

ある程度教師側には対象に対する解き方とか，考え方に前者のような無意識の直感があると考えてよろしいんじゃないでしょうか．そうすると，論理的な評価法がシステム側に書かれていればよい．しかし後者の直感は誤りも含んでいるわけですから，論理的な検証なしで正しく応答できる知的 CAI を作るのは非常に難しいだろうと思います．

竹内 私の勝手な考えですけども，たぶんここの4人の方が発表されたのは，全部知的なものを記号とし

て，ロジックで扱おうとしている立場なわけですね．ですから，その枠組みの中で直感をそのまま入れるということは，おそらく不可能であろうと思います．これはだから AI の問題として，何でもかんでも記号でやっていいのかと言ってる方もいらっしゃるわけですから，将来にわたって知的な CAI システムの中に，直感的な，論理的に明確にできないものが入らないとは言えないと思いますけども，そういうことを知的 CAI に取り込もうとしている方は，少なくともきょうの発表を伺ったかぎり，なかったように思います．

大槻 一言つけ加えさせていただきます．私個人の見解ですけれども，直感というのは経験からくるものではないかと思っています．経験のない人が，直感を得ることはまずないと思うんです．われわれは直感だと思ってますけども，山のような経験の上で何か物を言っているので，それが潜在意識にすぎないから直感だと思ってるのではないでしょうか．

ということは，今 AI で一番問題になっているように，常識推論とマシンラーニングの問題だと思います．非常に興味ある問題で，必ずしも方法論がないわけじゃないと思ってますけども，知的 CAI にそれを入れるには，まだ研究が必要ではないかと思っています．

今の問題でフロアからご意見ございますか．

フロアのコメント——今の直感というのは，大槻先生がご指摘のように，経験に基づいたものだと思うんですね．実際にそれを研究者の立場からみると，直感というものが，ドメインとか，教材とかいうものにどう置き換えられるかということを分析するのが研究者の立場であって，それを竹内さんが言われたように，記号表現なり，ニューラルネットなりに変えてやってシステム化するものだと思うんです．

大槻 それを渡辺先生がおっしゃったんですね．それじゃ，次の質問を．

質問者 2 松田先生がつくられた GEOMEX のことについてお伺いしたいんですけども，このシステムは幾何のセオレムプルーバとしても強力でしょうか．

松田 問題解決の能力に関する部分ですが，解法エキスパートは，汎用のプロダクションシステムです．今は三角形の合同証明に限定して，規則を与えています．補助線は考えていませんが，それ以外の問題でしたら解くことができます．

質問者 2 そうすると，入力可能な形としては，垂直とか平行とかいうのは許されてなくて，合同に関するものだけということでしょうか．

松田 キーワードと，それに関する解決規則を用意していますので，そのキーワード内ということです．

大槻 ほかにご質問をどうぞ．

質問者 3 最近知的という言葉をよくお使いになり，どういうことが知的かというのも分からないんですが，何と言うんでしょうか，数学とか，物理とか，わりときちっと決まってる分野でお使いになることが多いようですが，たとえば私が取り組んでおります歴史のようなもので，学生がどういう意図をもって何をしたいと思ってるかということをシステム側で認識するのは難しいと思うんですが，何か参考になるご意見がありましたらお伺いしたいんですが．

大槻 歴史のような，明確な回答をもたない分野で学習者の意図を認識する方法ですか．

質問者 3 はい，数式がきれいに決まってて答えが分かってるという問題は，何とかなるとは思うんですが，たとえば補助線をどう引くかといった問題にたくさんの解法があって，予測がつきにくいといったもの世界が閉じてないというんでしょうか，何言われるか分からないという先ほどの英会話の問題もあったんですが，そういったものを受け付けるためには，どういった方策を取ればよろしいんでしょうか．

大槻 これは渡辺先生の方法論で回答になりますね．

渡辺 教育の専門家として何かを教えようと思えば，教える領域を設定するはずです．それをはずれた学生を大事にしたいというのも教育的だとは思いますが，現状では，思わぬところへも展開していけるような教育をやろうという考えをおもちだったら，相当難しいものを志向することになると思います．

これに反して，たとえば歴史でも，細かく記述されている知識をシステムがもっていて，それで話題は大まかなところからスタートして，学生が細かいところへ突っ込んでいくとか，時代を変えて話題を進めていくとか，そういう機能はある程度達成できると思います．やはりシステムをつくるときに，それのもっている目標点というのをきちっとセットしておく必要があります．

大槻 ITS というのを今のアーキテクチャの範囲で考えますと，何を言うか分からないような叙述というのは，双方向主導対話は難しい．ユーザ主導でやりますと，これはデータベース的なもの，あるいは構造

をもったオペレーション，たとえば比較するという操作で年代比較，人物比較，状況比較のように使う方法はあると思うんですね．それから，システム主導でしたら，初めから意図をもって対話を誘導すればいいわけですから，これもできる．

だけど，双方向で何を言うか分からないものというのは，現在の技術では非常に難しいと思います．

ほかにご質問をどうぞ．

質問者 4 最初に松田先生にお願いします．

類似度を与えるというのは一般的に難しいと思うんです．類似度というメジャをどのように定義されておられるかというところに，まず一つ疑問がある．

それから，山本さんに質問があるんですが，ある場面を想定した英会話を教えるという場合に，素直に考えますと，すぐ物語理解の方法，たとえばスクリプトのような方法を使いたくなるんですが，ここでフレームを使われているということは，どういうところにメリットがあるのか，技法上のことで伺いたいのですが．

松田 きょうの話の中の枠組みでよろしいでしょうか．専門家の解法プロセスとの類似度という話です．

それは，教材構造知識ベースの中で表現するわけです．先ほど示しましたが，ばらばらに存在している前提が，もっとも集中している部分を取ってきて類似度を定義しています．幾何ということを考えますと，あるノードに近い前提というのは，図形の上で表現すると，大変近いところに位置しています．それで，こういった前提に多数決の原理を使って絞り込むことによって，ある程度の類似性を抽出できるということになります．

山本 最初のフレームに関するご質問ですが，われわれのところでは渡辺先生の格構造と似たフレーム構造を内部表現として採用しています．入力文をフレームに変換しまして，内部のフレームと動詞を中心にマッチングを取って会話を進めています．自由度の高い会話のための内部表現としては，動詞を中心としたフレームが都合がよいと考えています．

大槻 どうもありがとうございました．ほかにご質問ありますか．

質問者 5 この分野に関してはまったく素人なんで，変な質問になるかもしれません．人間教師の場合は，授業の間，学生の理解状態を把握するとき，話題に関する専門知識がなくて学生が誤りを犯す場合ももちろんあるんですけど，言い方と言いますか，表現上の誤りもあると思うんです．専門知識はもってるんですけど，表現の段階で言い方が間違ったため，間違いを犯したと仮定すると，先生の役割をコンピュータに任せた場合，そのコンピュータはどう対応するでしょうか．

大槻 入力の表現がおかしいという場合ですね．

質問者 はい．

大槻 今のご質問に対して教材が自然言語の場合は該当しないようですね．山本先生のは，自然言語そのものが対象ですから，入力表現が間違うということは，まさに解答そのものが間違っていることですから．それ以外の教科になるわけですね．

渡辺 その誤った表現というのは，自分は知ってるんだけども表現が正しくできないということですね．

山本さんと私はたぶん同じ立場だと思うんですけども，文法や意味構造に基づいて入力文を解析しておりますので，一応こういう誤りもありうるんじゃないかという判定はできるわけです．しかしそれらをはずれてしまう，システムに書かれてないやり方に対してもサポートするというのは難しい．

そこで現在の話題はどの場面に対応してるだろうかということをスクリプトなどを用いて解析し，対応する部分がいくつかあったらそれらを見せて，あなたの考え方はこれですか，と示すことはできますね．

竹内 学生の表現が悪いというのは，文法が間違っているとか，言葉の使い方が，標準的な使い方と違う場合と，解釈してよろしいんでしょうか．

質問者 5 はい．

竹内 そうしますと，まず，現実にシステムをつくっちゃいますと，文法的に日本語として正しくないものは，基本的には受け付けてくれないということになってしまいます．意味的にもおかしければ，計算機がもし正しく解釈できる知識しかもっていなければ，正しい解釈しか受け付けませんので，それも解釈できないということになってしまいます．

ただ，知的なシステムというのは，学生の間違ってもっている知識を推定するという機能を今もちはじめている，そういう研究もたくさんしてるわけです．

それと同じように，文法間違いとか，言葉の使い方についても，たった1人が一度しかしない間違いについては，これはどうしようもないですけども，大勢の人に観察される，たとえば特定の地域で使われる言葉の用法が，よその大部分の地域と違ってるというような場合，本来ならば間違いとみなすものも，だんだん

システムの中に取り込んでいくことは可能だと思います.

現状では,あらかじめシステムをつくる人がやらざるをえないでしょうけれども,将来文法とか語用について,先ほどお話がありましたけども,格フレームに対応して,名詞の意味が分類されているわけですけれども,そういうものを変更してみて,解釈できないかを試してみることによって,受け付けなかったものを受け付けるようにシステムが変わっていくという可能性はあると思います.

大槻 ほかの先生はいかがですか.松田先生何か.

松田 一つの考え方としまして,何を教えるかということを少し絞り込むのはどうかなと思います.

たとえば,正しい知識をもっているわけですから,ある教えたい知識があって,それは分かっているというのであれば,その誤った言い方というのは,インタフェースの部分で何とか処理してしまうという方法もあると思います.

そうではなく,正しい言い方までもちゃんと教授したいのであれば,誤った表現はどういうふうに誤っているかというのを内部にまで持ち込んで,どこがどう誤っているかを推論するようなメカニズムをもたせなければいけないと思います.

だから,どこまで教えるかによって解決策を考えるべきだと考えます.

大槻 どうもありがとうございました.

ほかにご質問をどうぞ.

質問者 6 ただいま私,CAI システムの設計を一生懸命やっている最中で,学習者モデル,教材知識,教授知識を設計するために苦労しておる最中です.専門家にインタビューして,問題をしゃべりながら解いていただいて,記録を残して解析するとか,いろいろやっておるんですけれども,設計を進めていきましてどうしてもつまずいてしまうのが,システムのすべての基盤になると思われる教材知識の内容をどう構築したらいいかというところです.

そこで,どの程度教材知識を充実させれば,インテリジェント性をもたせることができるか.誤り原因を同定して,学習者に与えた問題をシステム自体が解いて,学生が誤りを出したら,同じ誤りを生成できるような力をもたせるために,どの程度大きな知識ベースをつくればいいのかを,はかりかねております.

非常に低俗な質問になるかと思うんですけども,たとえば BOOK システムでは,Fortran の教授でしたらどの程度になっているのか,お聞きできれば,私どものシステム構築に非常に役に立つんですけれども.

竹内 どれだけ細かく知識を書くかということで,システムの能力はすべて決まってしまうわけです.引き算の例ですと教える順番というか,難易度に従ってまず知識を細分化するということですね.グループ分けをする.1桁の足し算にしても,片手の範囲でできる計算が一番簡単な足し算,次に,片手いっぱい,5に何かを足すというのが次のやさしい足し算,一番難しいのが,答えが両手を使わないと計算できない足し算というふうに,とにかく分けておくこと.

それから,初心者にある事柄を理解させるときには,ゼロから何か教えていくわけですけども,その教えるときに表れる概念というのは,すべて教材知識の中に盛り込むということはもちろん鉄則ですね.

Fortran の例ですと,割に簡単です.簡単というのは,Fortran は人間がつくったものですから,意味が正確に定義されていて,教えるときに必要な概念が比較的せまい範囲で閉じてしまいますから,その範囲内を知識として書いておけばいいわけです.

Fortran の学習をする人を対象にするのであれば,ゲートのオン・オフなんてレベルは必要ないですから,その辺はもう抽象化してしまう.あくまでも式の演算レベルで止めておいていいわけです.もしハードのほうまで勉強させる必要があれば,ゲートのオン・オフはちょっと行き過ぎかもしれませんけれども,たとえばレジスタを意識したレベルまで細かく知識として書いておかないといけないかもしれません.けれども,普通の Fortran であれば,そこまではいらないと思います.何をどこまで教えたいかということによって,どこまで詳細に書くのかが決まってくると思います.

Fortran の類は比較的簡単なんですが,そうでないもの,たとえば自然科学の教材知識を書くということになると,これは一概にどこまでというのは言えなくて,まさに先生の技量の発揮のしどころで,何を教えたいのかを先生がどう設定するかに,ひとえにかかってくると思います.

大槻 それ以外に必要なのは,やっぱりこの規則をどういう条件のもとで使うかという適用条件を付け加えなければいけないということと,なぜ適用できるかという理由を付け加えなければいけない.それをメタで解釈できるような機構にしないといけない.それが知的能力の一番本質じゃないかと思います.

じゃ，先ほど手が上がっていた方どうぞ．

質問者 7 竹内先生にお伺いしたいんですが，大型機とパソコンと両方に同じものをお入れになった．そのパソコンの移植について，さっき二つほど大型機には入ったけれども，パソコンには入らなかったというのがおありだったのですが，それは容量がパソコンで増えれば解決できるものなのか，もっと別な限界があるのかということをお伺いしたいと思います．

それから，松田先生にも同じことと，それから実際に初等幾何を中学生に使って，何か効果があったのかということと，二つお伺いしたいんですけど．

竹内 まず，パソコンの限界なんですが，これは今の感触としては，容量の問題です．

スピードに関しては，最新式のパソコンであれば，ワークステーションと比べても，そんなにスピードは変わりません．じゃ，ワークステーションを使ったらスピードが十分かというと，これは不十分です．でも，これはもう今のハードの限界ですから．ソフトでもっと工夫すべきところがあるんじゃないかと言われるとそうかもしれませんが，スピードだけを考えるのであれば，パソコンでも十分であろうと思います．

松田 私も同じ考えです．パソコンの限界として，やはりメモリの問題があります．Lisp とか Prolog とかいう言語は，たくさんのメモリを必要としますので，パソコンで大きなシステムを構築することは，困難だと思います．

それから，もともと実行速度を早くしようということを特に考えていませんでしたが，実用に耐える程度だという気はしています．

それから，学習効果をはかるという話ですが，システムの評価として何回かやっています．その結果，実際に学生の証明計画をシステムが認識して，それに対するコメントを与えるわけですから，その辺が大変おもしろいというコメントなどを学生から得ています．

竹内 ちょっと具体的に，パソコンで実験した例をあげますと，たとえば日本語の解析をするのに，「505 ひく 444 はいくつですか」のように単純な文を入力した場合ですと，一瞬ですみます．ところが，同じ意味にとれる文でも，解釈の仕方が何通りもあるような複雑な文を入力しますと，今のパソコンのシステムだと，全解探索をするのに，コンパイルしてても数分程度かかる場合もあります．

これは複雑な文も入力したいと，欲張ったことを言うからそうなるので，入力を単純な文に限れば，画期的に早くなります．結局のところ，何をどこまでやりたいのかが問題だと思います．

大槻 どうもありがとうございました．もう時間が過ぎておりますので，一応ここらで打ち切らせていただきたいと思いますが……．ごめんなさい，もう1人いらっしゃいましたね．どうぞ．

質問者 8 実際効果的な教授法ということについて，各知的 CAI がどの程度のことを考えていらっしゃるのか，お伺いしたいと思いまして．

大槻 効果的な教授法ですね．それを1人一言ずつコンクルーディングリマークスとしていただきたいと思います．実現しているのでも，将来計画でもいいです．

竹内 指導方法については，先ほど発表のときに方法をいくつか列挙したんですけれども，いろんな方法をうまく組み合わせるというのが，やはり一番だろうと思ってます．そのためには，システムがかなりいろんな情報をもっていないといけません．たとえば，先ほどの例をあげた中に，反例をつくるというのがありましたけども，反例が有効に働くためには，学生がその反例を見て間違っているということに気がついてくれないといけないわけで，そのためには，学生が知っている事柄をシステムが知っていて，その知っていることに矛盾する例をつくらないといけないわけです．そうすると，本来の問題解決に必要な知識以外に，成り立つべき性質に関する知識が要りますので，そういうものを全部入れておく，あるいは物理の問題であれば定性推論と組み合わせて，定量解析の指導をすると．そういうふうな組合せが必要になってくると思っております．

松田 効果的な教授法ということですが，やはり学習者の考えている事柄，または考えている延長上で教授を展開していくことが一番ではないかと思います．

たとえば，学習者が考えているのと，システムのメッセージとが全然違うのではよろしくないと思います．先ほども言いましたが，学生の考えている事柄と一致したメッセージが帰った場合に，大変興味を示しているという結果で出ているので，今後その辺をもう少し検討しなければいけないなと考えております．

山本 われわれのシステムですと，効果的な教授には，会話の臨場感を上げることが重要だと思います．そのために，より自然な会話を行えることはもちろんですが，それ以外に，画面表示とかレスポンスタイムを短くすることが重要だと思います．画面表示やレス

ポンスタイムを抜きにしては，やはり効果的な教授は考えられないと思っています．

渡辺 私自身は知的 CAI をやる方がどんどん増えてほしいというふうに思います．というのは，知的 CAI では今もっているシステムが，たとえば教授法に問題がある，そこをもっと充実したいとか，そういう転換が非常にやりやすい．私も Fortran とか Pascal とかの言語でプログラムを組んできましたけれども，Prolog は論理記述がものすごくやりやすいですね．

そのため，つくったプログラムが非常にうまく転用できる．確かにいろんな不十分な面もあると思うんですけれども，論理型プログラミング言語による知識表現の方法論を，うまく教育のコンピュータシステムに組み込んでいくほうに，ぜひ興味をもっていただきたいなというのが，私の実感であります．

大槻 どうもありがとうございました．

一応全部のお話をいただきましたので，最後に締めくくらないといけないわけでございますが，私聞いておりまして感じましたのは，すべてのものにはもちろん適用限界があるわけですね．たとえば，手続き的な言語で書いておりましたときには，つまり伝統的 CAI には皆さま方すでにもうご存じの適用限界があったわけです．

その適用限界をどうやって乗り越えようかということで，知的 CAI という考え方が出てきた．

そこでは現在までに幾つかの方法論が提案されています．たとえば先ほど議論された指導方略です．それから，知識表現では宣言型のほうがメタ推論が使いやすいというような方法論も出てきたわけですね．その方法に対しても，やっぱり手続き型で記述したときに限界があったのと同じような限界が当然あるわけです．

ですけど，その限界の範囲の中で，つまり記号処理による推論の中で知識情報処理のいろんなストラテジがあるわけですね．推論方法にしてもいろいろすでに出ているわけですね．たとえばディダクションだけじゃなしに，帰納推論とか，ある種のアブダクション，類推とか，連想とか，仮説推論とか，定性的推論とか，いろんな方法があるわけです．

そういうものを使って，今までできないと思っていたことを実現するのが，このインテリジェント・チュータリング・システムの目的ではないかと考えます．その適応限界の範囲の中でしたら，いろんな方法論があって，いろんなプログラムが書けるというわけですから，ぜひ皆さん挑戦して，新しいシステムを発表していただきたいというのが，このパネルセッションでの結論でございます．

来年 11 月 15 日 (1989 年) が，日本で開かれます国際会議の論文締切日でございます．今からちょうど 1 年余りございますので，今からでもシステムをおつくりになれる．完全にでき上がらなくても結構でございますから(笑)，きちっと設計をして，理論的なめどをつけて，論文を出していただければ，日本はこれぐらいやっているんだというのが示せます．いいお仕事をなさっている外国の研究者も招待したいと考えています．ぜひ皆さんが論文を出してくださいますよう期待しております．

では，これをもって終わらせていただきます．

どうもありがとうございました．(拍手)