解　説

知的 CAI における学習者モデル†

溝口 理一郎††　角所 収††

1. まえがき

従来の CAI では，教材の作成に主力が注がれ，そこに学習者モデルや教授知識などがすべて手続き的に表現されてきた．知的 CAI の一つの成果は，教育すべき教材の知識と制御知識が混然一体となっていた従来の教材を，教材知識，学習者モデル，教授戦略の三つのモジュールに分割してシステムを構築するという方法論を確立したところにある．これによって，システム構成の見通しがよくなったことばかりではなく，教授行為の本質が明確にされ，各モジュール，特に学習者モデルと教授戦略に関する研究の推進にも貢献している．

三つのモジュールの中で教授戦略モジュールはシステムの中枢であり，最も大切な部分であることはいうまでもない．しかし，教授戦略を立てるためには学習者の理解状態に関する情報が必要となる．人間相互のインタラクションが相手の理解状態を把握することなしには成立しないことからも分かるように，教師が学習者の状態を正確に把握して初めて適切な教育が可能となる．この意味で，学習者モデルは知的 CAI において高度な個別教育を行うための本質的な役割を担っているということができる[21]．このような認識のもとに，これまで学習者モデルに関する多くの研究がなされてきた[14),20),30]．本稿では，知的 CAI システムの研究の中で学習者モデルに焦点を当て，基本的な考え方を整理し，いくつかの学習者モデル表現の手法とその得失について述べる．

2. 学習者モデルの諸問題

本章では，学習者モデルを考える上で必要となる視点を整理するとともに，これまでに提案されてきたモデルを分類する．

2.1 学習者モデルの視点

（1） バグに対する考え方

学習者の理解状態をどこまで把握しようとするのかということはモデル構築における基本的な視点である．そもそもバグとは何か，そしてそれをコンピュータ上でどのように表現するのかという問題を考えなければならない．知識の不足，誤った理解，誤った知識をもつに至った過程，などバグを追求する深さの問題，言い替えれば，バグを表現する際に，それを再現するだけでよいのか，あるいはバグが発生した理由の説明もする必要があるのか，などということはモデル構築の本質的な問題である．

（2） モデル表現

モデル構築の方法には，あらかじめバグを整理しておき，学習者の誤りをそれらの組合せで表現する方法と，教材知識をバグの入り込む余地のない基本的なプリミティブに分解しておき，それらの組合せとして学習者の理解状態をそのまま表現する方法がある．前者は事前の準備が大変であるが，バグの概念を用いてトップダウン的にモデル化を行うので効率はよい．後者の方法にはバグの概念は存在せず，学習者の振舞いを素直にボトムアップ的な方法で表現しようとしている．

これに関連して問題となるのが表現プリミティブの大きさである．モデル表現の精密さと探索（モデル構築）の効率は相反する関係にあり，適切なトレードオフが必要である．

（3） 実行可能性

学習者モデルの検証を行うにはモデルが出す回答と学習者の回答とを比較するのが最も妥当な方法である．したがって，信頼性の高いモデルの構築にはモデルの実行可能性は不可欠である．さらに，学習者モデルを教育の過程で積極的に利用しようとするとき，学習者の振舞いをできるだけ正確に把握し，かつ予測することが望ましい．たとえば次に出すべき問題を決定するには，実際に学習者モデルで問題を解いてみて期待

† Student Model in Intelligent CAI by Riichiro MIZOGUCHI and Osamu KAKUSHO (I.S.I.R., Osaka University).
†† 大阪大学産業科学研究所

されるように誤るような問題を選択する必要がある．従来のように，単元や項目の習熟度を数値で評価するモデルではこのような木目の細かい出題はできない．

（4） ノイズ対策

学習者モデルは学習者の振舞い，具体的には問題に対する回答，をデータとして構築されるが，そのなかには次に挙げる三つの原因によりノイズが混入する．

① 表現言語が人間の理解や推論過程の近似にすぎないこと．

② 学習者が首尾一貫した振舞いをすることはなく，不注意による誤りなどが含まれること．

③ 学習が進むにつれて学習者の理解が変化すること．

このため，ノイズを含むデータからモデルを構築することを考えなければならない．

2.2 学習者モデルの分類

（1） オーバレイモデル[6]

楽観的にみれば，学習者は学習が進展するとともに知識が増加し，最終的には教師の知識と一致する．オーバレイモデルはこの考えに基づき，学習者の知識を教師の部分集合として捉えるモデルである．

（2） 差異モデル[3]

学習者と教師の解法の差異に注目して，学習者が未習得の知識を検出するという立場に立つモデルである．能力的にはオーバレイモデルと同等である．

（3） バグのモデル

オーバレイモデルや差異モデルではバグ（誤り）という概念が希薄であった．この型のモデルはバグをモデル化することに重点が置かれている．

（a） 数え上げ型

あらかじめさまざまな誤答を解析し，誤った回答手順を数え上げてネットワークの形で表現するモデルである．バギーモデル[1]とも呼ばれる．

（b） 再構成型

① バグの概念あり[4),24)]

事前に学習者の多くの誤答を解析して誤った基本的手続きを整理し，正しい基本知識とともにモジュール構造で蓄積しておき，それらの組合せとして学習者の理解状態をモデル化する．

② バグの概念なし[10),16)]

誤った知識に限定するのではなく，学習者が用いると考えられる最も詳細なレベルの基本手続きを整理し，それらの組合せとして学習者の理解状態を表現する．したがって，システムは何が誤った知識かは知らない．

（c） 原因分析型[2),17),29)]

上に述べたモデルは，深くてもバグを再現するところまでで，バグが発生した原因にまでさかのぼってモデル化するものはなかった．この方法では学習過程の認知モデルに基づきバグ発生過程をもモデル化する．

3. 学習者モデル構築

本章では前節の分類に基づいていくつかの学習者モデルについて述べる．

3.1 オーバレイモデル

この方法による学習者モデル構築は学習者が習得した項目にフラッグを立てるだけでよく，モデルの構築がきわめて容易であるという特徴がある．しかし，これでは学習者の知識の不足はモデル化できても誤った知識のモデル化ができない．このように，オーバレイモデルは記述能力が低いにもかかわらず，その使いやすさゆえに歴史的には最もよく使われてきた[5),7)]．また，その能力の強化も試みられており，岡本らは差異モデルと結合したより実用的な方法を提案し，微分演算の教育を行うシステムで用いている[19)]．

3.2 バグのモデル

3.2.1 数え上げ型

バギーモデルは Brown らによって作られた BUGGY[1)] システムで提案された．彼らは引算における小学生の解答例を分析して，基本的な手続きを同定し（正しい手続きと，誤った手続きの両方が含まれる）それらを手続きネットワークと呼ばれる表現形式で構成した．たとえば，「必ず大きい数から小さい数を引く」や「上位の桁から借りてもそこから1を引かない」などのよくみられる誤りが基本手続きとしてネットワークのノードに登録されている．BUGGY はバグのモデルを初めて構築した点で高く評価されており，その後の研究に大きな影響を与えた．しかしながら，バギーモデルにはモデルの表現能力があらかじめ作られたネットワークに直接依存し，それに限定されるという欠点がある．また，例題からモデルを同定する機能はないため学習者の診断ではなく，間違った解答を学習者に提示して学習者にその誤りを発見させるという逆転した方法で利用された．

3.2.2 再構成型

（1） バグの概念あり

BUGGY におけるバグに対する基本的な考え方を継承しているが，モデルの表現能力を高くするために，各基本手続きをネットワークではなく，できるだ

け自由度が高い形で保持するという方法がとられている．バギーモデルより表現の柔軟性に富むが，事前に行うバグ分析の負担が大きいという共通の問題が残されている．オーバレイモデルについでよく用いられてきたモデル化技法である[4),24)~26)].

（a） DEBUGGY[4)]

BUGGY を学習者モデルの構成が可能なように拡張したものが DEBUGGY である．DEBUGGY は BUGGY で解析した基本手続きを組み合わせて生成・テスト法によりモデルを構築する．学習者の解答が一度にすべて(複数個)与えられることから，off-line 型と呼ばれている．また，複数の誤りが組み合わされて生じるバグまで考慮すると探索空間が非常に大きくなるという問題が生じる．そこで DEBUGGY では探索空間を縮小するために，複雑なバグを構成する基本的なバグはそれらおのおのが単独でも検出できるという仮定を置いている．

まず初めに，単独のバグときわめて頻度の高いバグの組合せからなる仮説空間を作成し，支持する証拠(解答)の数を考慮して仮説を縮小する．その後，仮説を結合してよりよく解答を説明する仮説を求める．組合せの数の上限を超えても満足のいく仮説が得られなければモデル構築に失敗し，停止する．

DEBUGGY では Coercions (強制) と呼ばれるノイズ対策がなされている．この方法では，与えられた解答がよくみられる不注意な誤りかどうかを，事前に準備された情報に基づき判断し，そのような解答をモデル化の対象から除外する．もちろん，不注意な誤りと誤解に基づく誤りとを明確に分離することは困難であり，Coercions も不十分なものでしかない．なお，DEBUGGY を実際の問題で評価した結果，約 2/3 の学習者の診断には成功したが，残りの学習者については診断が不能であったという結果が得られている．

（b） IDEBUGGY[4)]

IDEBUGGY は解答をインタラクティブに一つずつ受取りながらモデルの構築を行うことから，on-line 型と呼ばれている．構築の手続きは，①競合する仮説を分離するために有効な問題の出題，②複雑なバグをもつ仮説の生成，③以前中断された仮説の復活，などのいくつかの独立したモジュールに分割され，それらを効率よく起動するスケジューラの制御のもとにモデル構築が行われる．

各バグにはそれを同定するために有効となる問題の経験的な生成法が記述されており，それに従って出題される．ノイズ対策としては，怪しい場合には数値だけが異なる類似した問題を出すことによって，不注意によるものか，誤解によるものかをある程度区別できるように工夫がなされている．

対話的に学習者モデルを構築することから，IDEBUGGY では探索を考慮する仮説の数を制限している．さらに，バグの出現頻度順に優先順位を付け，単純なバグから複雑なものへと仮説を構成していく．システム構成としては DEBUGGY より進化しているが，性能評価に関する報告は見られない．

（c） パーターベーション法[25)]

バグは正しい知識になんらかの変形（パーターベーション）が加わったものと考えることができる．バグを解析して変形の一般的な形態をオペレータとして整理して蓄積しておき，それらを正しい知識に適用することによって学習者の振舞いの説明を試みるのがパーターベーション法である．パーターベーションオペレータの適用の制御や適用の対象としての正しい知識の選択などに未知の部分が残されているが，正しい知識と少数のオペレータだけで学習者モデルが構成できるという性質は魅力的である．竹内らは教材に依存しないパーターベーションオペレータと依存するオペレータとの2種類のオペレータを導入した学習者モデル構築法を提案している[26)]．

（2） バグの概念なし

学習者モデルの構築は，学習者の解答を例題としてそれらを説明する一般的な理論（モデル）を作ることであることを考えれば，人工知能研究における例題からの学習や帰納推論が有効なことが分かる．しかも，学習者の理解状態をそのまま表現することを試みれば，事前にバグの解析をしてバグのカタログを作成しておく必要がないという利点がある．表現の基本として，詳細な手続きが用いられるため表現力は十分高いが，効率の点で問題が残されている．一方，この方法はボトムアップ的な方法であるのでノイズの影響を受けやすく，ノイズ対策が必要となる．

（a） ACM[16)]

Langley らは ACM (Automatic Cognitive Modeling) という，人工知能で研究されてきた学習理論を用いてプロダクションルールで学習者モデルを構築する方式を提案している．まず，問題空間を設定するために状態の記述と，問題の解決に必要な基本オペレータを定義する．引算におけるオペレータの例を図-1 に示す．プロダクションルールは実行部にオペレータ

10を加える（数，行，列）：その行とその列が指定する数を取り出し，それに10を加えて置き換える．
1を引く（数，行，列）：その行とその列が指定する数をとりだし，それから1を引いて置き換える．
左へ移動（列）：現在注目している列から一つ左へ移動する．
差を取る（数1，数2，列）：おなじ列から二つ数を取り出し，その差をその列の答えとして書く．

図-1 引算における基本オペレータ

を，条件部にそのオペレータの適用可能な状況を記述することによって構成される．学習の対象は学習者が適用したオペレータの系列の同定を通した各オペレータの適用条件である．

学習者モデルの推論は次の三つのステップに従って行われる．

① 人手による基本オペレータと問題空間の設定．

② 学習者が用いたオペレータ系列の候補の数え上げ．

③ オペレータの適用条件の推論．

ACMにおける探索は基本的には総当たり，すなわち学習者の回答を説明するすべてのパス（オペレータの系列）を探索するので，効率を向上させるための経験則を導入しているが，教材に依存したad hocのものであった．そこで，新しく考えられているのが，オペレータの認知的に妥当な適用に関するルール，DPF (Diagnostic Path Finder) である[30]．ありそうにない適用を除外する規則と競合する適用を評価する規則とに分かれている．まだ開発途上であるが，完成が期待される．

（b） 帰納推論法

筆者らは学習者モデル記述言語にPrologを採用し，学習者モデルをPrologのプログラムとして構成する方式を提案している[13]．したがって，学習者の誤解はプログラムの虫（バグ）として表現される．見方をかえれば，学習者モデルの構築はPrologプログラムの合成問題となるが，合成アルゴリズムとしては，ShapiroのMIS (Model Inference System)[23]を採用している．このことにより，CAIシステムをMISを用いた帰納推論による学習者モデル構築と，Prologプログラムの虫取りとしての教育という二つの枠組みの結合体として定式化することができる．

MISは有限個の正の事実と負の事実を入力とし，それらを証明するPrologのプログラムを合成するシステムである．プログラムが負の事実を証明してしまうとき，そのプログラムは強すぎると言われ，正の事実の証明に失敗すると弱すぎると言われる．どちらの場合においてもPDS (Program Diagnosis System) を用いて原因が探索される．強すぎるプログラムを弱めるときには誤りの原因となったホーン節を取り除き，弱すぎるプログラムを強めるときには，原因となったゴールをカバーすべきホーン節を合成する精密化作用子がそれぞれに必要となる．いずれの場合にも，必要が生じれば事実の真理値に関する質問が行われるが，教育的にみて妥当なものであり，積極的に問い合わせることによって学習者モデルを構築していくという方法論を提供している．

学習者モデルの表現力を強化するために，筆者らは{true, unknown, false, fail} の4つの真理値をもつ言語SMDLとそれを用いた帰納推論法の開発を行っている[10),12)]．true, unknown, falseはそれぞれ学習者がある事実を正しいと思っている，知らない，間違いと思っていることを示す．そしてfailはシステムが学習者の理解状態を知らないことを意味する．この拡張によりモデル表現が自然で強力になっている（これは，たとえば学習者モデルの初期状態が空のとき，「システムは学習者が何も知らないと思っている」と考えるより，「システムは学習者の状態を知らない」と解釈するほうが自然であることからも分かる）．

さて，学習者の応答すべてを説明するモデルの構築を行っている筆者らの方法では，ノイズ対策として誤ったデータからの帰納推論，すなわち非単調な帰納推論が必要となる．非単調推論では，真理値が確定しない命題（仮定）と確定しているものとを区別して管理し，矛盾が発生した際に，その解消に貢献し得る仮定を効率良く求める機構が用意される．このようなことを行うために考えられている機構は一般にTMS (Truth Maintenance System) と呼ばれている．TMSに基づく学習者モデル構築システムでは，モデル構築の途中経過の情報をTMSに与え，矛盾（モデル構築の失敗）が起きればそれをTMSに伝える．TMSは推論過程を解析することによって，矛盾の解消に必要な仮定を同定し，その真偽に関する信念を返す．より具体的にいえば，TMSは学習者モデル構築システムが矛盾解消のために訂正すべき学習者の応答の候補を教えるわけである．現在，DoyleのTMS[8]とde KleerのATMS[15]を導入して学習者モデル構築方式の強化を行っている[10),11)]．

帰納推論に基づく方法は，これまで述べてきたモデル構築法のなかでは，ACMと並んで表現力の意味で

最も優れているが，その汎用性の高さの代償として，効率が悪いという問題がある．これに対しては，汎用の帰納推論方式のなかにバグに関する教材に依存する経験則を記述できる機構を用意することによって，汎用性を損わない効率の改善策が施されている．

3.2.3 原因分析型

これまで述べてきた再構成型のモデルでは，バグの再現はできてもバグが発生した原因を説明することはできないことはすでに述べた．バグの問題を考察するとき，学習者が学習を進めていく過程を考慮にいれるべきであることはある意味では当然であり，その考察はよいモデルを構築するために有効な知見を与えてくれる．バグが発生する原因の究明はバグの問題を深く理解するために必須のものであるが，それは学習者が知識を習得していく過程の認知モデルとも関連する興味ある課題でもある．

（1） REPAIR 理論[2]

学習者は問題が解けなくなると，勝手な考えに基づいてその場しのぎの解決を試みるため，一貫した誤りがみられない場合がある．したがって，そのような現象の表面的なモデル化を試みることは無意味であり，再構成型の方法ではモデルを作ることができない．このような状況に対しても有効なモデル理論を構成するには学習者の学習過程のモデル化を検討しなければならない．

図-2 に引算の例を示す．Brown と Vanlehn はこのような学習者の振舞いを考察することによって，REPAIR 理論と呼ばれる次に示す二つの過程からなるバグの生成理論（モデル）を提案している．

① Impasse と呼ばれる困難な状況の発生

② それを解決するための修復操作 (REPAIR)

Impasse の発生の原因となった手続きは核手続き (Core procedure) と呼ばれるが，このような考えに立てば，現象に一貫性がないのは修復の操作がその場しのぎであるためであり，真の原因としてのバグは，REPAIR が必要となる状況を作った原因としての核手続きであることが分かる．核手続きは一般には誤った手続きや正しい知識の欠落が考えられるが，REPAIR 理論では正しい知識の欠落 (Deletion principle) だけが考慮されている．

手続き的知識は GAO グラフと呼ばれる一般化された AND/OR グラフで表現される．ノードにはゴールを表すプロダクションルールが記述されており，処理の流れを陽に制御する．Impasse に出会うと制御はプロダクションシステムから局所的な問題解決器に渡され，REPAIR が行われる．REPAIR は無制限に行われるものではなく，その時点の実行状態に関連する範囲に限定される小規模で一時的な処理とみなされる．REPAIR の認知的な正当性を保証するために，具体的には次の三つの修復原理が採用されている．

① 教材独立性：汎用の問題解決器でサポートできる範囲の処理，たとえば処理のスキップ，後戻り，類似したオペレータとの交換などの操作に限定される．

② 局所性：大規模な修復は行われないと仮定している．

③ 独立性：Impasse の原因と修復操作とは独立であると仮定されている．

BUGGY で整理されたバグの生成を試みることによって REPAIR 理論の評価が行われたが，かなり多くのバグの生成に失敗し，結果は芳しくなかった．この原因はいうまでもなく，Impasse 発生の原因となった核手続きの生成モデルが不十分であることにある．問題解決につまずく原因には正しい知識が欠落していることだけではなく，誤った知識を適用してしまったことも考えられるが，REPAIR 理論では前者のみが考慮されているため学習者の振舞いを正しくモデル化することができない．

（2） STEP 理論と SIERRA

Vanlehn らは REPAIR 理論の欠点を補うために，STEP 理論[28]と呼ばれる理論を提案した．STEP 理

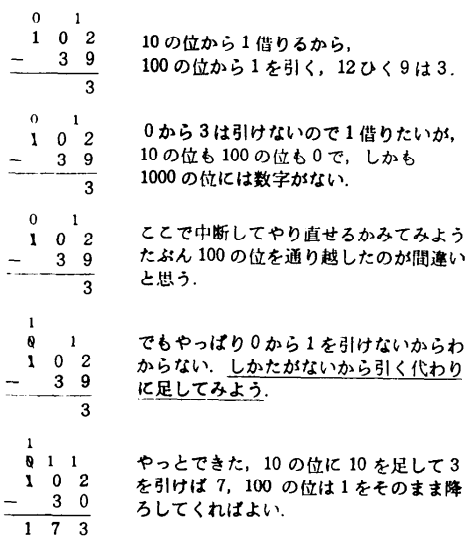

図-2 バグ生成過程の例（下線部は REPAIR 操作を表す）[27]

論は，手続き的技能を課題の進展に従って段階的に習得していく学習過程をモデル化したものであり，核手続きの生成を説明するための基礎を提供するものである．STEP 理論の本質は技能の学習が例題からの帰納推論により行われるとするところにあるが，帰納する，すなわち一般化する際に誤りが生じると考え，核手続きの生成を説明している．

REPAIR 理論と STEP 理論を統合したものが SIERRA[29]である．学習モデルは課題ごとに帰納推論により学習し，その過程で核手続きを生成する．問題解決部は REPAIR 理論に従って修復を行う．そして，解答を DEBUGGY に送り診断を行う．

Impasse の生成に関しては次の三つの仮定がなされている．

① 過度の一般化
② 過度の特殊化
③ 知識の欠如

SIERRA は REPAIR 理論を拡張したものであることから，バグ生成の性能は（まだ不完全ではあるが）従来のものより優れていることが実際の例を用いて検証されている．

（3） プロセスモデル[17]

これまで述べてきた方法の多くは，学習者の誤りは「誤った知識を正しく使う」ことによって生じると仮定されていた．Matz[17] は高校程度の代数を対象にして学習者の振舞いを分析し，誤りの多くは知識そのものではなく，むしろ「すでに獲得した正しい知識を新しい状況において適用する際に生じる」とみなしたほうが自然であるという考え方（プロセスモデル）を提案している．プロセスモデルでは誤りは次に示す三つの場合に分類されている．

① 補完手法の誤った選択

人間はだれでも見知らぬ問題に直面したときには，すでに所有している知識とその問題とのギャップを埋めるための補完法を考える．補完が成功すれば問題の解決に成功するが，失敗することも多い．実際多くの誤りはこの補完の失敗が原因となっている．Matz は代表的な補完法として線形補完と一般化をあげている．線形補完とは，分配則のように部分構造を独立に扱って操作を施した後，その結果を組み合わせるような操作において，適用してはならない対象に対してもそのまま素直に適用してしまう誤りを意味する．たとえば，

$A(B+C)=AB+AC \rightarrow A(B*C)=AB*AC$

$\sqrt{A*B}=\sqrt{A}*\sqrt{B} \rightarrow \sqrt{A+B}=\sqrt{A}+\sqrt{B}$

$\frac{B+C}{A}=B/A+C/A \rightarrow \frac{A}{B+C}=A/B+A/C$

などがある．一方，一般化の誤りには

$(X-3)(X-4)=0$
$X-3=0$ or $X-4=0$
$X=3$ or $X=4$

という問題を一般化して

$(X-A)(X-B)=K$
$X-A=K$ or $X-B=K$
$X=K+A$ or $X=K+B$

という間違いを犯し，

$(X-3)(X-4)=2$

という問題では $X=5$ or $X=6$ と答えてしまう．

② 新しい概念の獲得の失敗による基本知識の不足

算数から代数へと教材が進展した場合には，まったく新しい概念を獲得しなければならず，それにつまずくと基本知識が不十分になってしまう．たとえば，算数では数字の連接は桁の位置を考慮した加算を意味していたが，代数では積を意味する．したがって，$4x=48$ という方程式を解かせると $x=8$ という誤答が得られることがある．また，符号の概念は，算数では左辺の計算をして結果を右辺に求めるという意味であるが，代数では両辺の等価性を主張する形式的なものであり，この理解の不足によっても誤りが生じる．

③ 手続き実行の失敗

この型の誤りはさらに計画の失敗と処理の失敗の二つに分類されて論じられているが，上の二つの誤りに比べてプロセス理論らしさに欠けるので省略する．

4. 学習者モデルの今後の課題

これまで学習者モデル構築のさまざまな研究を眺めてきたが，多くの研究者の努力にもかかわらずいくつかの問題が残されている．2. で述べたノイズ対策はその代表的なものであるが，すでに触れたのでここでは他の課題について考察する．

（1） 認知過程を考慮したモデル化

学習者モデルの構築は認知過程にまで考察を深めた研究が盛んになりつつある．我が国においても学習者の問題解決過程の認知モデルの追求を指向した学習者モデル表現の手法が，中村，平島[9),18)]らによって考察されている．学習者モデルを構築するという立場からみれば，学習者の理解状態から学習の過程まで含めてできるだけ正確にモデル化したいと考えるのは自然な

流れではある．しかし，その実現には従来の方法よりさらに高度なモデル化技法が必要とされる．バグそのものを深く理解することは教育を進める上で重要であるが，その発生過程のモデル構築をすることは容易ではない．モデル構築技法の高度化も考慮した研究が望まれる．

（2） 学習者からの情報の獲得

これまでの研究の多くは学習者の解答だけからモデルの構築を試みていた．一般に，少ない情報からのモデル化は困難であるが，研究としては興味深く，また学習者に対する介入が少なくなるという利点がある．しかしながら，現在多くの研究者が対象としているような高度なモデルを構築するには，利用する情報が少なすぎるように思われる．学習者の学習過程の流れを損わない程度の自然な問いかけは可能であるはずであり，学習者自身にもっと多くを語らせることによって，モデルの構築の効率向上に役立てる工夫が必要であると思われる[22]．

（3） 教授戦略との連携

学習者モデルは，教育を進める上で教授戦略モジュールが必要とする情報を提供することを目的としているにもかかわらず，多くの研究は教授戦略の考察とはある程度無関係に行われてきた．Self[22]も指摘するように，教授戦略で有効に利用することができないような精密なモデル化は不用である．モデル構築の可能性の検討に加えて，教授戦略におけるその利用法を考慮した総合的な研究が必要であろう．

（4） モデル構築の運用

モデル構築の終了の判定（原理的には不可能）は困難な問題であるので厳密な追求をしないことにしても，構築を終了して教育に取りかかるタイミングの決定は重要である．言い換えると，いくつの問題からモデルを構築するかということである．誤答がみつかる度にモデルを構築し，誤解を正すことも考えられるが，一つの例題から学習者モデルを構築するのは一般に危険である．一方，モデルの正確さを重視すれば多くの例題が必要となるが，誤りを長い間放置することは好ましくない．学習者モデル構築モードとバグ取り(教育)モードの区別があからさまに見えるのは避けるべきである．モデル構築と教育の間の制御の移行が適応的で円滑に行えるような機構の開発が望まれる．

5. むすび

学習者モデルに関する数々の研究の結果，精密なモデル構築の可能性が明らかにされてきた．しかしながら，4. で述べたように解決しなければならない課題が多く残されているのも現実である．学習者のモデルを構築することは，とりもなおさず人間の理解のモデルを構築することにほかならない．研究者が設定した形式的な記述言語を用いて，不可解なる人間の理解の様子をモデル化するという壮大な目標が達成されるにはまだ多くの年月が必要であると思われる．しかし教育に必要なモデルは完全なモデルではなく，あくまでも近似で十分なはずであり，その時点で利用可能な教育技術が必要とする情報を提供することができればよいのである．モデル化技法の高度化だけではなく，モデルの適正な精度，必要な情報とそのための学習者との対話，必要なコンピューティングパワーなどを総合的に考慮して，各時点における最適なモデル構築法を見いだす努力も必要であろう．

参考文献

1) Brown, J. S. et al.: Diagnostic Models for Procedural Bugs in Basic Mathematical Skills, Cognitive Science, Vol. 2, No. 2, pp. 155-191 (1978).
2) Brown, J. S. et al.: Repair Theory: A Generative Theory of Bugs in Procedural Skills, Cognitive Science, Vol. 4, No. 4, pp. 379-426 (1980).
3) Burton, R. R. et al.: An Investigation of Computer Coaching for Informal Learning Activities, Int. J. Man-machine Studies, Vol. 11, No. 11, pp. 5-24 (1979).
4) Burton, R. R.: Diagnosing Bugs in a Simple Procedural Skill, Sleeman, D. H. et al. eds., Intelligent Tutoring Systems, Academic Press, London, pp. 157-183 (1982).
5) Carbonel, J. R.: AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction, IEEE Trans., Vol. MMS-11, No. 4, pp. 190-202 (1970).
6) Carr, B. et al.: Overlays: A Theory of Modeling for Computer Aided Instruction, MIT AI Memo 406 (1977).
7) Clancey, W. J.: Tutoring Rules for Guiding a Case Method Dialogue, Int. J. Man-machine Studies, Vol. 11, No. 1, pp. 25-49 (1979).
8) Doyle, J.: A Truth Maintenance System, Artificial Intelligence, Vol. 12, pp. 231-272 (1979).
9) 平島他：意図と方略の概念を用いた学生の誤りのプロセスモデルの提案とその実現，信学技報，Vol. 88, No. 7, ET 88-1, pp. 65-72 (1988).
10) 池田他：知的 CAI のための知識表現と帰納推

論，信学技報，COMP 86-79 (1987).
11) 池田他：仮説型モデル推論システム，Proc. of the Logic Programming Conference '88, pp. 47-56 (1988).
12) Ikeda, M. et al.: Design of General Framework for ITS, Proc. of ITS '88, pp. 82-89 (1988).
13) 河合他：論理プログラミングと帰納推論による汎用的CAIシステム，情報処理学会論文誌，Vol. 26, No. 6, pp. 1089-1096 (1985).
14) 木村他：知的CAIシステムにおける学習者モデルの動向，CAI 学会誌，Vol. 4, No. 3, pp. 5-14 (1985).
15) de Kleer, J.: An Assumption-Based TMS, Artificial Intelligence, Vol. 28, pp. 127-162 (1986).
16) Langley, P. et al.: Automated Cognitive Modeling, Proc. of National Conference, on Artificial Intelligence, Austin, Texas, pp. 193-197 (1984),
17) Matz, M.: Toward a Process Model for High School Algebra, Sleeman, D. H. et al. eds., Intelligent Tutornig Systems, Academic Press, London, pp. 25-50 (1982).
18) 中村他：類推ドメイン原理を用いた学生モデルの構築手法について，信学技報，Vol. 88, No. 7, TE 88-1, pp. 65-72 (1988).
19) 岡本他：知的 CAI のための教授世界知識の表現とその推論の方法，電子情報通信学会論文誌，Vol. J 70-D, No. 12, pp. 2658-2667 (1987).
20) 岡本：知的 CAI, 電子情報通信学会誌，Vol. 71, No. 4, pp. 384-390 (1988).
21) 大槻：高度個別教育における知識情報処理—ITSの機能と振舞い—，人工知能学会誌，Vol. 1, No. 2, pp. 196-201 (1986).
22) Self, J.: Bypassing the Intractable Problem of Student Modelling, Proc. of ITS '88, pp. 18-24 (1988).
23) Shapiro, E.: Algorithmic Program Debbuging, MIT Press (1982).
24) Sleeman, D. H. et al.: Modeling Students' Problem Solving, Artificial Intelligence, Vol. 16, pp. 171-187 (1981).
25) Sleeman, D. H.: Inferring (Mal) Rules from Pupiles' Protocols, Proc. of European Conference on Artificial Intelligence, Orsay, France, pp. 160-164 (1982).
26) 竹内他：摂動法による学習者モデル形成と教授知識について，情報処理学会論文誌，Vol. 28, No. 1, pp. 54-63 (1987).
27) VanLehn, K.: On the Representation of Procedures in REPAIR Theory, CIS Report 16, XEROX PARC (1981).
28) VanLehn, K.: Human Procedural Skill Acquisition: Theory, Model and Psychological Validation, Proc. of the National Conference on Artificial Intelligence, Washington, D. C., pp. 420-423 (1983).
29) VanLehn, K.: Learning one Subprocedure per Lesson, Artificial Intelligence, Vol. 31, No. 1, pp. 1-40 (1987).
30) Wenger, E.: Artificial Intelligence and Tutoring Systems, Morgan Kaufmann Pub. Inc. (1987).

(昭和 63 年 7 月 14 日受付)