知識工学と人工知能 54－6
（１９８７・９・１８）

学生の誤りの発生過程のモデル化と教育への利用

平島 宗　中村祐一　上原邦昭　豊田順一
大阪大学産業科学研究所

本報告では、学生の誤りを知識の誤りとして捉えるのではなく、問題解決のプロセスにおける知識の用い方の誤りとして捉え、この考え方に基づく誤りのモデル化を実現するために、問題解決のプロセスを意図の設定と意図をより詳細な意図へと展開するための手続きの適用とにより表現する。このモデル表現では、学生の誤りは、設定された意図に対する手続きの誤った適用としてモデル化される。本モデルでは、バグと不安定で一貫しない誤りは同一の表現により表わされるが、誤りに対する指導の過程でこれらの誤りを区別することにより適切な指導が可能となっている。本報告で述べるモデル化の対象領域は中学初年度程度の式の計算とする。

A New Generative Theory of Errors and their Modeling for Education

Tukasa HIRASHIMA, Yuuichi NAKAMURA, Kuniaki UEHARA, Jun'ichi TOYODA
OSAKA University
The Institute of Scientific and Industrial research
8-1, Mihogaoka, Ibaraki, Osaka 567 JAPAN

This paper develops a unifying account for some observed systematic errors in arithmetic and early algebra problem solving. It proposes that a error arises from incorrect adaptation of a known procedure. This proposal idealizes an individual's problem solving behavior as a process model. The notion of the process model is formally defined herein as a trace of the student's mental operations in deciding which steps to perform. Once the process model is constructed, it can be analyzed to reveal the incorrect adaptation of a known procedure. This analysis can be used to remedy for the student's error.

1. 序論

知的ＣＡＩシステムにおいて学生の誤りに対し適切な指導を行うためには、学生の誤りを正確にモデル化することが必要である。数の計算や初等代数等の手続き的知識を教育する知的ＣＡＩシステムでは、これまで多くの場合、バグモデルが用いられてきた[1][2]。バグモデルは、システム内に用意された、学生が持つ可能性のある誤った知識（バグ）を用いて、学生の誤りを表現するモデルである。一般に、バグモデルは学生の犯した誤りを再現することだけを目的としており、学生がバグを生成する過程については考慮されていない。この問題を解決する一つの方法として、学生の持つバグの生成過程をモデル化しようとする研究が行われている。主な研究としては repair theory[3] や Matz モデル[4]等があるが、これらのモデルは、不安定で一貫しない誤りに対処できないという欠点があった。

本報告では、従来のモデル化手法の欠点について考察し、この欠点を解決するための方法として、意図と手続きに基づくモデル化手法を提案する。本モデル化手法では、学生の誤りを知識の誤りとして捉えるのではなく、問題解決のプロセスにおける知識の用い方の誤りとして捉える。この考え方に基づくモデル化を実現するために、問題解決のプロセスを意図の設定と意図をより詳細な意図へと展開するための手続きの適用とにより表現している[5]。意図とは学生が問題を解くための目標であり、手続きとは学生が問題を解くための具体的な手段である。このモデル表現を用いると、学生の誤りは、設定された意図に対する手続きの誤った適用としてモデル化される。本モデルでは、バグと不安定で一貫しない誤りは同一の表現により表わされるが、指導の過程でこれらの誤りを区別することにより適切な指導が可能となっている。

本報告で述べるモデル化の対象領域は中学初年度程度の式の計算とする。

2. これまでのモデル化手法の問題点

学生の誤りをモデル化する手法として最も盛んに用いられているのがバグモデルである。バグモデルは、学生の誤りをシステム内に用意されたバグを用いて学生の誤りを表現するモデルである。バグモデルを用いて知的ＣＡＩシステムを構築する場合、バグごとに指導法を用意しなければならないという問題点がある。これは、バグモデルが学生の誤りを再現することだけを目的としており、学生がバグを生成する過程を考慮していないためである。例えば、学生が手続きＰを用いるべきところで、誤った手続きＰ’を用いたとする。バグモデルを用いると、学生が手続きＰの代わりに手続きＰ’を持っていることをモデル化することは可能であるが、学生が手続きＰ’を生成する過程についての情報は何も得られない。すなわち、バグモデルを用いても指導のための有益な情報を得ることはできない。

この問題を解決する一つの方法として、誤った手続きＰ’がいかに生成されたかをモデル化し、そのモデルからバグを修正するための情報を得ることが挙げられる。誤った手続きの生成過程のモデル化の研究の主なものとしては repair theory と Matz モデルがある。ここでは、repair theory について検討してみる。repair theory では学生の持つ誤った手続きは、手続きが書き換えられることにより生成されるとしている。書き換えのための方略を間に合わせ方略という。学生が問題解決過程で既知の手続きでは対処できない「行き詰まり状態」に陥った場合、学生は間に合わせ方略を用いて自分の知っている手続きから「行き詰まり状態」を解決するための手続きを生成する。このときに生成された手続きが誤っている場合、バグが生成されたことなる。repair theory は多桁減算の領域に存在するバグについて、妥当な生成過程の説明を行っており、その考え方は有効であると考えられる。ところが、誤りに関する最近の研究により学生の誤りの多くが不安定であり、明確にバグといえる一貫した誤りが少ないことがわかってきた[6][7]。repair theory は学生の誤りがバグにより説明できることを前提とし、バグの生成過程のモデル化を目的としたものであるため、不安定な一貫しない誤りに対して妥当な説明が行えない。こうした点を考慮して、著者等は、学生の誤りを知識の誤りとして捉えるのではなく、問題解決のプロセスにおける知識の用い方の誤りとして捉えることにより、不安定で一貫しない誤りを捉えることのできるモデル化手法の提案を行う。

3. 誤りのモデル化

3.1 誤りの分類

バグモデルでは、学生の誤りを一貫性を持ったバグによる誤りと、ランダムに発生するスリップによる誤りに分類している。ところが、最近の誤りに関する研究によれば、学生の誤りをバグとスリップにより説明することが困難であることがわかってきた。そこで著者等は、手続きの用い方に注目し、学生の誤りは問題解決のプロセスにおいて適切でない手続きが用いられるために発生すると考えた。この考えに基づいて学生の誤りを次のように分類する。

・手続きの誤った適用による誤り
・誤った手続きの使用による誤り

・スリップ

○手続きの誤った適用による誤り

誤りの大部分を占めるのは手続きの誤った適用による誤りである。この誤りは、正しく用いられる条件を持つ手続きが、適切でない条件で用いられたために発生する誤りである。例えば

$(a+b)^2 = a^2+b^2$

は式を展開するときに分配法則を用いてしまったための誤りとすることができる。手続き「分配法則」自体は誤っておらず、有用な手続きであるといえる。例えば、$(a\times b)^2 = a^2\times b^2$ は手続き「分配法則」が適切に用いられた例である。従来誤った手続きの使用によるとされてき誤りの多くは、この手続きの誤った適用による誤りとして説明することができる。この誤りは、人間にとって解釈の容易な誤りである。

この誤りの捉え方では、バグによる一貫した誤りと不安定で一貫しない誤りの違いは頻度だけであり、二つの誤りは、同様に手続きの誤った適用による誤りとして説明できる。

手続きの誤った適用が発生する原因は、手続きの用い方がわかっていないためであると考えられる。そして、手続きの用い方を理解していなくても手続きを用いることができることが誤りの発生する根本的な原因と考えられる。

○誤った手続きの使用による誤り

誤った手続きとは手続き自体が誤っており、正しく用いようがない手続きを用いたことによる誤りである。この誤りは、未知の手続きを用いているため捉えるのが困難である。例えば

$(-5)+0=5$

の誤りは 0 との和をとると符号が変わるという誤った手続きを用いていると考えられるが、明らかではない。このような誤った手続きは教師や教科書により学生に教授されたとは考えられないから、学生が独自に獲得した誤りであり、学生の知識獲得の一面を表すもので、単なる手続きの適用の誤りとは違うものと考えられる。このため、人間にとって解釈のしにくい誤りである。幸いなことに誤りの事例に含まれるこの種の誤りは少ない。

○スリップ

スリップはランダムに発生する誤りで、単なる計算間違いである。

ここでは誤りの大部分を占める手続きの誤った適用による誤りについてのモデル化を行う。

3.2 手続きの誤った適用による誤り

ここでは学生の誤りを手続きの誤った適用による誤りとして捉えることにより、バグを用いずに誤りが説明できることを示す[9]。この説明は、誤りが一貫した誤りであるか、あるいは不安定で一貫していない誤りであるかに関係なく、誤りに対する十分な説明になっている。

(例1)

$3\times 4 = 7$

この誤りは 3×4 を評価するときに加算の手続きが用いられたための誤りと考えることができる。加算の手続きが和の式に用いられれば、式は(例1-1)のように正しく変形される。

(例1-1)

$3+4 = 7$

このことから、(例1)の誤りの原因は、加算の手続き自体にあるのではなく、3×4 を評価するときに加算の手続きが用いられたことにあるといえる。

(例2)

$(a+b)^2 = a^2+b^2$

この誤りは $(a+b)^2$ を展開するときに手続きとして分配法則が用いられたための誤りと考えられる。分配法則が $(a\times b)^2$ や $(a+b)\times 2$ の式に用いられれば、式は(例2-1)や(例2-2)のように正しく変形される。

(例2-1)

$(a\times b)^2 = a^2\times b^2$

(例2-2)

$(a+b)\times 2 = a\times 2+b\times 2$

このことから、(例2)の誤りの原因は、分配法則という手続き自体にあるのではなく、分配法則が $(a+b)^2$ の式を展開するときに用いられたことにあるといえる。

(例3)

$0.3+0.4 = 0.07$

この誤りは答えの小数点以下の桁数を決めるときに、手続きとして、与えられた問題の各項の小数点以下の桁数を加算し、得られた数を答えの小数点以下の桁数とする、積算の場合に用いられるべき手続きが用いられたための誤りと考えられる。この手続き自体が小数点以下の数を持った数の積算の式に用いられれば、式は(例3-1)のように正しく変形される。

(例3-1)

$0.3\times 0.4 = 0.12$

このことから、(例3)の誤りの原因は、この手続き自体にあるのではなく、手続きが $0.3+0.4$ の式に用いられたことにあるといえる。

(例1)~(例3)では、誤りを手続き自体の誤りではなく、手続きの用い方の誤りとして説明している。さらに、誤って用いられた手続きは全て正しく用いることができる条件を持つ。このような誤りを手続きの誤った適用による誤りとする。

本システムでは、誤りを手続きの用い方の誤りとしてモデル化するために、手続きと適用条件の分離を行っている。適用条件は手続きが適用できる条件を示す

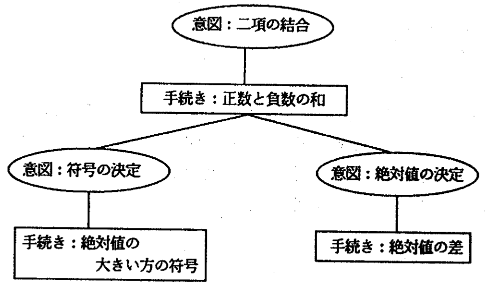

図１　(+8)+(-2) = +6 を行う問題解決のプロセス

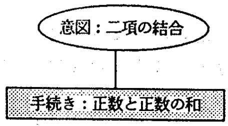

図２　(+8)+(-2) = +10 を行う誤った問題解決のプロセス

もので、誤った適用条件を持てば適切でない問題に対して手続きを適用できる。このため、手続きの用い方の誤りは、誤った適用条件により表現できる。さらに、手続きの誤った適用を制限するために手続きの意図を用いている。意図は手続きの用いられる目的を表すもので、手続きが用いられるときには意図が存在することが前提条件となる。意図は例１では評価、例２では展開、例３では小数点の決定にあたり、用いられる手続きはそれぞれの意図を満たしている。

3.3　問題解決のプロセスの表現

本節では問題解決のプロセスの表現について述べる。

式変形を行うためには、式変形の目標を設定することが必要となるが、この目標を意図と呼ぶ。意図が設定されるとその意図を達成するために意図の段階的詳細化が行われる。意図の段階的詳細化は、手続きを用いて意図をより詳細な意図へと展開する過程を繰り返すことにより行われる。意図が最も詳細な意図に展開されると、プリミティブな手続きにより実行される。プリミティブな手続きはオペレータであり、プリミティブな手続きを実行することは、このオペレータを用いて入力から出力を計算することである。以後はプリミティブな手続きをオペレータと呼ぶ。

図１は

（＋８）＋（－２）　＝　＋６

を行う正しい問題解決のプロセスである。この例を用いて、問題解決のプロセスの説明を行う。まず、問題に対してどのような式変形を行うかというトップレベルの意図が設定される。トップレベルの意図は入力としての問題の形式と出力としての解の形式により決まる。（＋８）＋（－２）　＝　＋６　の場合、入力は（＋８）と（－２）の二つの項と‘＋’の演算子から構成されているので、入力の形式は［項，演算子，項］と表現できる。出力は（＋６）の一つの項から構成されているので、出力の形式は「項」と表現できる。入力の形式［項，演算子，項］から出力の形式「項」を得る式変形を行うための意図として「二項の結合」が設定される。

意図が設定されると、その意図を実行するための手続きを選択することが必要となる。選択される手続きは、意図の持つ入出力形式と同じ入出力形式を持つ手続きである。意図「二項の結合」の意図を実行する手続きは、入力の形式が［項，演算子，項］であり、出力の形式が［項］でなければならない。この入出力の形式を満たす手続きとしては「正数と正数の和」，「正数と負数の積」等が存在する。選択できる手続きが複数存在する場合、手続きの持つ適用条件と、意図が式変形を行う対象である副問題との照合により選択される手続きを一つに決める。適用条件は手続きの適用可能な条件を表す。意図が式変形を行う対象である副問題は、意図に対する具体的な入力である。図１においては、意図に対する具体的な入力は　（＋８）＋（－２）　である。このとき、手続き「正数と負数の和」を選択の候補としてみる。手続き「正数と負数の和」の適用条件は、二項が正数と負数であり、演算子が和であることである。この適用条件は［項（正数），項（負数）、演算子（和）］と表現できる。意図に対する具体的な入力からは、項は正数と負数であり、演算子が和であることがわかる。これは手続き「正数と負数の和」の適用条件を満たしている。このため、手続き「正数と負数の和」が意図に対して選択される。

手続き「正数と負数の和」により、意図「二項の結合」は「符号の決定」と「絶対値の決定」という２つの意図に展開される。

意図「符号の決定」は、各項の符号と絶対値と演算子の入力［［符号，絶対値］，演算子，［符号，絶対値］］から符号の出力［符号］を得ることを目標とする意図である。この意図に対しては、二項の符号が異符号であり、かつ演算子が和であるときに適用可能な手続き「絶対値の大きい方の符号」が選択される。この手続きはオペレータであるので、入力に対して出力をただちに得る。この場合、オペレータは［［＋，８］，＋，［－，２］］の入力に対して［＋］を出力する。意図「絶対値の決定」に対して選択される手続き「絶対値の差」もオペレータであり、入力［［＋，８］，＋，［－，２］］に対して［６］を出力する。

次に、学生の誤った問題解決のプロセスについての表現を説明するために、図２と図３に誤りの例を示す。図２は手続きの適用を誤った例であり、図３はオペレータの適用を誤った例である。

図２は学生が

（＋８）＋（－２）　＝　＋１０

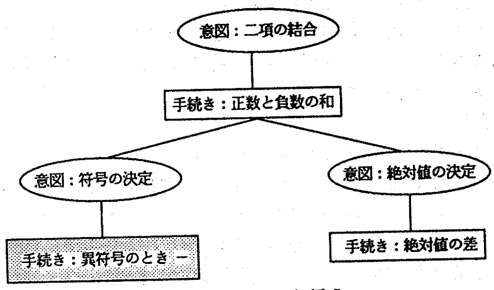

図３　(+8)+(-2) = -6を行う
誤った問題解決のプロセス

のような誤りを犯した場合の問題解決のプロセスである。この問題解決では、意図「二項の結合」に対する手続きとして「正数と正数との和」が選択されている。この手続きの正しい適用条件は、二つの項が正数と正数であり、演算子が和であることである。正しい適用条件は［第一項（正数），第二項（正数），演算子（和）］と表せる。意図が対象としている副問題は（＋８）＋（－２）　であるから、そこから項と演算子がなにであるかを抽出すれば、二つの項が正数と負数であり、演算子が和であることがわかる。これは［第一項（正数），第二項（負数），演算子（和）］のように表せる。この条件は手続き「正数と正数との和」の誤った適用条件である。　（＋８）＋（－２）　＝　＋１０　の誤りは、手続き「正数と正数との和」が誤った適用条件［第一項（正数），第二項（負数），演算子（和）］に従って用いられた結果として説明できる。

図３は学生が
（＋８）＋（－２）　＝　－６
のような誤りを犯した場合の問題解決のプロセスである。この問題解決では、意図「二項の結合」に対しては適切な手続き「正負の数の和」を選択したが、意図「符号の決定」に対して適切でない手続き「異符号で－」を選択している。手続きは「異符号で－」オペレータである。オペレータ「異符号で－」の正しい適用条件は、二つの項の符号が異符号であり、演算子が積であることである。これは［第一項（正数），第二項（負数），演算子（積）］と表せる。このオペレータが適用された副問題は、（＋８）＋（－２）　であるから、この副問題から項と演算子の条件を抽出すると、二つの項の符号が異符号であり、演算子が和であることがわかる。この条件は［第一項（正数），第二項（負数），演算子（和）］と表せる。この誤りは手続き「異符号で－」を誤った適用条件［第一項（正数），第二項（負数），演算子（和）］で適用したための誤りと説明できる。

以上のように、このモデル表現を用いると、手続きの適用の誤りとオペレータの適用の誤りを適用条件の誤りとして表現できる。

3.4　誤った問題解決のプロセスの同定

3.3 節で述べたように、学生の誤りは、意図に対しての手続きの誤った適用により発生すると考えられる。そこで、誤りのモデルとして、誤った手続きが用いられた誤った問題解決のプロセスを生成する。ここでは、図４を用いて誤った問題解決のプロセスの生成について述べる。

本モデル化手法では、正しい問題解決のプロセスで用いられた適切な手続きを適切でない手続きに置き換えることにより、学生の誤りを再現する誤った問題解決過程を生成する。例えば、（＋８）＋（－２）を解く場合、意図「二項の結合」に対して適用される手続きは「正数と負数の和」であるが、これを「正数と正数の和」や「正数と負数の積」に置き換えることに

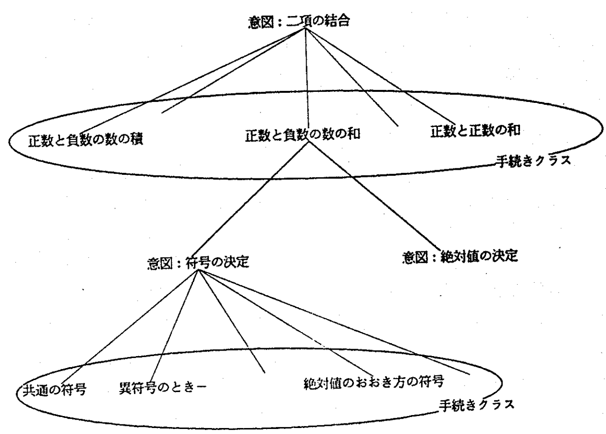

図４　誤った問題解決のプロセスの同定

より、誤りのモデル化が行える。この際、置き換えを行うための手続きの候補が問題となるが、図4に示すように、これらの候補は手続きクラスとしてまとめられている。手続きクラスとは、同じ入出力形式を持つ手続きの集合である。例えば、意図「二項の結合」についての手続きクラスには、「正数と正数の和」、「正数と負数の和」、「正数と負数の積」、等の手続きがある。

（例）

（＋8）＋（－2）　＝　＋10

の場合の誤った問題解決のプロセスの同定を例として、図4を用いて、誤った問題解決のプロセスの同定を行う。まず、意図「二項の結合」に対する手続きの置き換えを行う。置き換えられる手続きは手続きクラスにまとめられているので、このクラスの中から置き換える手続きを選択する。例えば、手続き「正数と正数の和」が選択されたとする。この手続きにより、式（＋8）＋（－2）は正数と正数の和として計算され、出力として　＋10　が得られる。以上のようにして問題解決のプロセスは同定される。

3.5 誤った問題解決のプロセスの診断

誤った問題解決のプロセスのモデルは、学生の誤りをモデル化したものである。このモデルを診断することにより、学生の誤りに対する指導に必要となる情報を抽出することができる。

本モデルを診断することにより、バグモデルでも抽出できた「誤って選択された手続き」と「選択されるべきであった手続き」に加えて、「手続きの選択の誤りが発生した意図」と「誤った適用条件」と「考慮されなかった条件」が抽出される。

「手続きの選択の誤りが発生した意図」は、学生がなにをしようとしたときに手続きの選択を誤ったのかを表わすので、この「手続きの選択の誤りが発生した意図」を用いれば、学生の誤りが問題解決のプロセスのどの部分で発生したかを指摘することができる。

「誤った適用条件」は、誤って選択された手続きが適用されたときの条件を表わす。手続きの選択の誤りが発生したのは、この「誤った適用条件」に基づいて手続きを選択したためと説明できる。

「考慮されなかった条件」は、誤って選択された手続きの正しい適用条件と誤った適用条件の差である。つまり、正しい適用条件と誤った適用条件とで食い違っている条件である。誤った適用条件を正しい適用条件に修正してやるには、この「考慮されなかった条件」について修正してやればよい。

「手続きの選択の誤りが発生した意図」と「誤って選択された手続き」と「選択されるべきであった手続き」は、誤った問題解決のプロセスを生成する過程で得られる。次に、「誤った適用条件」と「考慮されなかった条件」の抽出について述べる。

3.5.1 誤った適用条件の抽出

手続きの用い方の誤りは、適用条件の誤りとして表現できる。誤った適用条件は問題より正しい適用条件に基づいて抽出される。

（例）

（＋8）＋（－2）＝　＋10

問題に対して誤って用いられた手続き「正数と正数の和」の正しい適用条件は［第一項（正数），第二項（正数），演算子（和）］である。正しい適用条件の各成分に基づいて、誤った適用条件を問題より抽出する。誤った適用条件は［第一項（正数），第二項（負数），演算子（和）］となる。誤りは、手続き「正数と正数の和」が誤った適用条件［第一項（正数），第二項（負数），演算子（和）］で用いられたために発生したと説明できる。

（例）

（＋8）＋（－2）＝－6

問題に対して誤って用いられた手続き「異符号のとき－」の正しい適用条件は［第一項（正数），第二項（負数），演算子（積）］である。正しい適用条件の各要素に基づいて、誤った適用条件を問題より抽出する。誤った適用条件は［第一項（正数），第二項（負数），演算子（和）］となる。誤りは手続き「異符号のとき－」が誤った適用条件［第一項（正数），第二項（負数），演算子（和）］で用いられたために発生したと説明できる。

3.5.2 考慮されなかった条件の抽出

手続きの用い方の誤りの多くは、適用条件の一部が照合されなかったり欠けていたために発生すると考えられる。そこで正しい適用条件と誤った適用条件の差をとることにより、手続きを選択する際に考慮されなかった条件を抽出できる。誤った適用条件の内、考慮されなかった条件を修正すれば正しい適用条件となる。このため、考慮されなかった条件が手続きが誤って選択された原因とすることができる。手続きの誤った選択を修正するための指導は、考慮されなかった条件に重点をおいて行われる。

（例）

（＋8）＋（－2）＝　＋10

手続き「正数と正数の和」が選択される際に考慮されなかった条件は正しい適用条件［第一項（正数），第二項（正数），演算子（和）］と誤った適用条件［第一項（正数），第二項（負数），演算子（和）］の差により求められる。考慮されなかった条件は［［第二項，［正数，負数］］］である。問題の第二項が負数であるのに、第二項が正数でなければならない手続き「正数と正数の和」を用いたと説明できる。

（例）

（＋8）＋（－2）　＝－6

手続き「異符号のとき－」が選択される際に考慮さ

れなかった条件は、正しい適用条件［第一項（正数），第二項（負数），演算子（積）］と誤った適用条件［第一項（正数），第二項（負数），演算子（和）］の差により求められる。正しい適用条件では演算子は積であるが、誤った適用条件では演算子は和である。このため、考慮されなかった条件は［［演算子，［積，和］］］と表わせる。問題の演算子が和であるのに演算子が積でなければならない手続き「異符号のとき－」が選択されたと説明できる。

4．誤りに対する指導

4.1 基本方針

誤りのモデルは学生の誤りを表現したものであり、誤りに対する指導に利用される。本章では本モデル化手法により生成されたモデルを用いた誤りに対する指導について述べる。ここで用いられる指導法は、学生の誤りを直接指摘するのではなく、ヒントを与えることにより学生自身に誤りを気付かせる方法である。また、誤って用いられた手続きについても、誤った適用条件を指摘し適切な適用条件を与えることにより、誤って用いられた手続きの正しい用い方を指導し、手続が再び誤って用いられることのないようにする。

4.2 誤りの分類

本モデル化手法でモデル化された誤りは、全て手続きと手続きの誤った適用条件で表されるが、指導に対する学生の反応により次の三つに分類される。

（1）問題の理解の誤り
（2）適用条件の照合の誤り
（3）適用条件の不備による誤り

（1）の誤りは問題を誤って解釈したため、与えられた問題とは別の問題として解いてしまったための誤りである。例えば、（＋８）＋（－２） を （＋８）＋（＋２） と誤って理解してしまい、＋１０ と解答するような誤りである。（２）の誤りは正しい適用条件を持っているにもかかわらず、正しく照合しなかったための誤りである。例えば、（＋８）＋（－２） を行うときに、演算子が和であることをわかっていながら積のときの符号の決め方の手続きを用いて、－６ と答えるような場合の誤りで、誤りに対する直接的な指導を行わなくても、ヒントを与えることにより、学生が自分で誤りに気付き修正できる誤りである。（３）の誤りは適用条件の記述自体の誤りで、適用条件の欠落と条件自体の誤りがある。この誤りはヒントを与えても修正されない誤りである。

これらの誤りの診断と指導は（１）から（３）へと段階的に行われる。すなわち、システムは学生の誤りを（１）の誤りであると想定して指導を行い、それでも誤りが修正されない場合には、（２）および（３）の誤りを想定して指導を行う。

（１）、（２）は不安定な一貫しない誤りであり、（３）は固定的な一貫した誤りである。これらの誤りは本モデルでは区別して表現されていないが、モデルに基づく指導の過程で各誤りを分類することができる。ここでは、不安定な一貫しない誤りとバグによる誤りの違いは、誤りに対する指導のどの時点で誤りが修正されるかだけである。

4.3 誤りに対する指導

学生の誤りに対する指導は、まず、問題の理解の誤りであると仮定して行われる。この指導で誤りが修正されれば誤りは問題の理解の誤りである。問題の理解の誤りに対する指導では誤りが修正されない場合、適用条件に関する誤りとして指導を行う。この指導において、ヒントを与えるだけで誤りが修正されれば、適用条件の照合の誤りである。ヒントだけでは誤りが修正されない場合は、適用条件の不備による誤りである。

4.3.1 問題の理解の誤りに対する指導

誤りを問題の理解の誤りと仮定した場合の指導について述べる。例えば、（＋８）＋（－２）＝ ＋１０ のような誤りの場合、学生は問題を （＋８）＋（＋２） と理解していると仮定する。この誤りに対する指導は、学生に与えられた問題を確認させることにより行われる。そのために、学生が誤って理解していると思われる部分を指摘してやる。指摘は、モデルより得られる、考慮されなかった条件を用いて行われる。この問題において考慮されなかった条件は第二項の正負で、誤りを犯した学生は、第二項の負数であるにもかかわらず、第二項を正数として扱っている。そこで、学生に問題の第二項の正負を確認させる。問題を確認することにより誤りが修正されれば、その誤りは問題の理解の誤りである。問題の理解の誤りである場合、問題と学生が誤って理解した問題との違いを確認させるために、学生が誤って解いたと思われる問題を示す。この場合では、（＋８）＋（＋２）を与えればよい。

（例）
（＋８）＋（－２） ＝ ＋１０ に対する指導
（１）第二項に注目しなさい。
（２）第二項の正負に注目しなさい。
（３）第二項は負数であって、正数ではない。
（４）あなたは（＋８）＋（＋２）を計算しました。問題は （＋８）＋（－２） です。第二項の正負が違います。

4.3.2 適用条件の照合及び不備による誤りに対する指導

適用条件の誤りに対する指導は、誤った適用条件に

関するヒントを与えることにより行われる。ヒントを与えるのは、学生が正しく解けると仮定しているからである。まず、誤った手続きの適用が行われた意図を指摘する。意図の指摘により学生が考慮しなければならない範囲は大きく縮小される。次に、どのような条件が考慮されなかったために発生した誤りであるかを指摘する。さらに、考慮しなければならない条件を具体的に指摘する。ここまでの指導は、誤りを直接指摘していないので、誤りが修正されれば適用条件の照合の誤りと診断できる。誤りに対する間接的な指摘で誤りが修正されないとき、直接的に誤りの指摘を行う。直接的な指摘は、「誤用された手続き」、「誤った適用条件」、「誤用された手続きの正しい適用条件」、「適切な手続き」により行われる。

直接的な誤りの指摘の際に行われる説明はそのまま誤りの一般的な説明である。そこで、間接的な指導で誤りが修正された場合でも、誤りが修正された後に、直接的な誤りの指摘を用いて誤りの一般的な説明を行う。

（例）

（＋８）＋（－２） ＝ －６ に対する指導

（１）符号を決めるときに誤りました。

（２）演算子に注意して符号を決めなさい。

（３）演算子は和です。

（４）手続き「異符号のとき－」を誤って用いたための誤りです。この手続きは演算子が和である場合に用いてはいけません。演算子が積のときに用いるべき手続きです。演算子が和のときは手続き「絶対値の大きい方の符号」を用いるのが適切です。

（１）～（３）の指導で誤りが修正されれば適用条件の照合の誤りである。（４）の指導が必要なときは適用条件の不備による誤りである。

5. むすび

本報告では、学生の誤りを手続き自体の誤りではなく、手続きの用い方の誤りとして捉え、誤りのモデル化を行った。さらにモデルを用いた誤りに対する指導について述べた。本モデル化手法とモデルを用いた誤りに対する指導法の特徴は、次のようになる。

１）問題解決のプロセスに注目し、誤りを手続きの適用の誤りとして捉えることにより、本モデル化手法では誤りが一貫したものであるかどうかに関係なく、誤りをモデル化できる。

２）手続きと適用条件を用いることにより、本モデルでは誤りの発生を誤った適用条件に基づく手続きの適用により説明することができる。

３）問題解決のプロセスのモデルに意図を導入することにより、本モデルでは学生が何をしようとしたときに誤ったのかを指摘することができる。

４）誤りに対する指導を段階的に行うことにより、一貫した誤りに対しても不安定な一貫しない誤りに対しても適切な指導ができ、さらに指導の過程で誤りの分類が行える。

今後の課題は、モデル化手法の精練、知的ＣＡＩシステムとしてのインプリメント、モデルを用いた教育効果の評価である。

参考文献

[1] Burton, R. R.: Diagnosing bugs in a simple procedural skill, in Sleeman, D. & Brown, J. S. (Ed.), Intelligent Tutoring Systems, pp.157-183, Academic Press, London, (1982).

[2] Young, R. M., O'Shea, T. : Errors in Children's Subtraction, Cognitive Science, 5, pp.153-177, (1981).

[3] Brown, J. S., VanLehn, K.: Repair Theory: A Generative Theory of Bugs in Procedural Skills. Cognitive Science , 4, pp.379-426, (1980).

[4] Matz, M.: Toward a process model for high school algebra errors, in Sleeman, D. & Brown, J. S. (Ed.), Intelligent Turtoring Systems, pp.25-50, Academic Press, London, (1982).

[5] 平島，中村，上原，豊田： 誤った知識の形成過程のモデル化，第１回人工知能学会全国大会，9-7, pp.473-476, (1987).

[6] VanLehn, K.: Bugs are not Enough: Empirical Studies of Bugs, Impasses and Repairs in procedural Skills. The Jurnal of Mathematicl Behavior, 3. pp.3-71, (1982).

[7] 森田，佐藤，星野： 多桁減算における誤りの研究：オルタナティブな見方の提案，千葉大学 教育工学研究，第８号，pp.41-62, (1987).

[8] 石谷，小川：数学誤算誤答の事例研究 （代数），明治図書，(1960).

[9] Davis, Bobert B.: Learning Mathemaics: The Cognitive Science Approach to Mathematics Education, Croom Helm, London & Sydney, (1984).