

拡張オーバレイモデルに基づくCAIシステム ～教授ロジックと教材の作成事例～

小野寺 直樹† 馬場 純子‡ 山本 洋介†
中溝 昌佳† 古宮 誠一†

†芝浦工業大学大学院 〒337-8570 埼玉県さいたま市見沼区深作 307
‡芝浦工業大学情報工学科 〒337-8570 埼玉県さいたま市見沼区深作 307
E-mail: †{ m103195, m104106, m103198, skomiya }@sic.shibaura-it.ac.jp
‡101081@sic.shibaura-it.ac.jp

あらまし 近年、インターネットの普及により、Web 環境下で稼働する学習支援システムに期待が寄せられている。しかし、実際、商用で利用されている CAI システムのほとんどは、教授ロジックが直線型になっているので学習効率のよいシステムとは言えない。そこで著者らは、システムが学習者の理解状況(何処が学習者の不得意分野であるか)を把握することにより、学習者が苦手としている分野へ向って教材画面がどんどん分岐して行くようなシステムの構築を目指している。そこでは、学習者の理解状況をシステムが把握するために、既存のオーバレイモデルを拡張した新しい学習者モデルを採用している。これにより、教材作成の容易さを維持したまま学習者の理解状況を細かく把握することができる。このシステムがどのように学習者の理解状況を把握し、その結果、どのような教授ロジックを展開するかを、具体例を挙げて明らかにしている。

また、著者らは、分岐型教授ロジックに基づく教材の作成を容易にするために、教授ロジックを rule-base で実現する方式を採用している。

キーワード 拡張オーバレイモデル、学習者モデル、分岐型教授ロジック、WBT 型 CAI

An Enhanced-Overlay-Model-based CAI System: An Example of Instruction Logic & Teaching Materials

Naoki ONODERA † Jyunko BABA ‡ Yousuke YAMAMOTO †
Akiyoshi NAKAMIZO † Seiichi KOMIYA †

†Graduate school of Shibaura Institute of Technology,
307 Fukasaku, Minuma-ku, Saitama-city, Saitama,337-8570 Japan
‡Shibaura Institute of Technology
307 Fukasaku, Minuma-ku, Saitama-city, Saitama,337-8570 Japan
E-mail: †{m103195, m104106, m103198, skomiya}@sic.shibaura-it.ac.jp
‡101081@sic.shibaura-it.ac.jp

Abstract Recently, it is required to develop a WBT-CAI system to be able to learn efficiently due to the spread of Internet. However, since the order of giving a chain of text and decision frame does not change, if ever the learner solves the given problems, or the learner can not learn efficiently. Therefore, the authors propose a new student model which improved the existent overlay model. The new model is easy to create the teaching materials, and it is able to grasp the situation of a learner's understanding in detail. This paper clarifies how this system grasp the situation of a learner's understanding and what kind of instruction logic is deployed through showing a concrete example of the instruction logic.

By the way, the authors facilitate to provide the teaching materials according to the instruction logic through employing a rule-base CAI system.

1. はじめに

近年、インターネット(特に Web)の普及により、Web を用いた学習支援システムに期待が寄せられている。これは、Web 上に学習支援システムを構築すれば、誰もがいつでも何処でも学習支援システムを利用できるからであり、時間的および地理的な条件によらず、このシステムを

利用するすべての学習者に対して同一の教材を同時に提供できるからである。このような状況を踏まえ、Web 環境下で稼働する学習支援システム、WBT (Web-Based-Training)型 CAI (Computer Assisted Instruction) システムの研究開発が行われている。しかし、実際に商用として利用されている CAI システムのほとんどは教材の提示順序(教授ロジック)が誰に対しても同じ(直線型)

なので、学習効率のよいシステムとは言えない。この問題を解決するには、学習者の解答が正解であるか否かで教材の提示順序が動的に変化する、分岐型[6][8][13]の教授ロジックを採用する必要がある[7]。

本研究では、分岐型教授ロジックの考えをさらに押し進めて、システムが学習者の理解状況(何処が学習者の不得意分野であるか)を把握することにより、学習者が苦手としている分野へ向って教材画面がとどンドン分岐して行くようなシステムの構築を目指している。このようなシステムを構築するには、システムが学習者の理解状況を把握できなければならない。このため、学習者の理解状況をシステムが理解する指標として学習者モデルを使う。しかし、学習者モデルとしてオーバレイモデルを採用すると、他の学習者モデルよりも教材作成は容易になるが、学習者の理解状況の把握には適さないという問題がある。また、オーバレイモデル以外の学習者モデルを採用すると、学習者の理解状況の把握には適しているが教材作成は難しくなるという問題がある。この問題を解決するために、既存のオーバレイモデルを拡張した新しい学習者モデル(拡張オーバレイモデルと呼ぶ)を提案する。これにより、教材作成の容易さを維持したままで学習者の理解状況(学習者の不得意分野)の把握が可能となる。本論文では、拡張オーバレイモデルの採用により、学習者の理解状況(何処が学習者の不得意分野であるか)をどのようにして把握し、それによりどのように教授ロジックを展開するかを明らかにするとともに、教授ロジックの具体例を挙げて提案した方法の有効性を示す。

ところで、分岐型の教授ロジックに基づく CAI システムは学習効率が良いにも拘わらず、ほとんど構築されていない。その理由は教材作成が難しいからである。本システムでは、分岐型教授ロジックに基づく教材の作成を容易にするために、教授ロジックを rule-base で実現する方式を採用している。なお、本稿では学習対象の事例として、ソフトウェア技術者を習得するのに必要不可欠な Java 言語の文法習得を目的とする教材を採り上げる。

最後に、本論文の構成を示す。第 2 章では、直線型教授ロジックと分岐型教授ロジックを比較し、本研究で分岐型教授ロジックを採用した理由について述べる。第 3 章では、学習者モデルの例と、本研究が目指す教授ロジックを実現するために、拡張オーバレイモデルという新しい学習者モデルを提案し、拡張オーバレイモデルの概要を具体例を用いて説明する。第 4 章では、本システムで採用する教授ロジックを詳述する。第 5 章では、CAI システムの構築方法である、AFO-CAI[13]と rule-based CAI を比較し、教授ロジックに合わせて教材を具体的に組み立てる作業を容易にするために、本システムでは後者を採用することを述べる。第 6 章では、教材の具体例を挙げて、提案した拡張オーバレイモデルとそれに基づく教授ロジックの有効性を示す。第 7 章では、本論文のまとめを述べる。

2. 直線型と分岐型の教授ロジック

教授ロジックには 2 種類のタイプがある。1 つ目は、学習者の解答が正解であるなしに拘わらず、次々に提示される教材画面が常に同じものになる直線型[6][8]の教授

ロジック(Linear programming)で、Burroughs Frederick Skinner による Teaching Machine において初めて示された[10][11]。このため、Skinner 型教授ロジックとも呼ばれる。[6][8]直線型教授ロジックは学習者の理解状況を全く考慮していないので、明らかに学習効率が悪い。Norman A. Crowder[4]はこの欠点を解決するために、学習者の解答が正解であるか否かで、次々に提示される教材画面が動的に変化する、分岐型の教授ロジック(Branching program または Intrinsic program[4])を提唱した[8]。このため、Crowder 型教授ロジックとも呼ばれる。分岐型教授ロジックは学習者の理解状況を考慮した初めての教授ロジックであるという点で高く評価されている。学習効率の悪いにも拘わらず、市場に出回っている CAI システムのほとんどは、直線型教授ロジックを採用している。その理由は、分岐型教授ロジックは学習効率はよいが、教授ロジックに合わせて教材を具体的に組み立てる作業が困難だからである。

本研究では、分岐型教授ロジックをさらに推し進め、学習者の理解状況(何処が学習者の不得意分野であるか)を把握することによって、次々に提示される教材画面が動的に変化するような教授ロジックの CAI システムの構築を目指している。

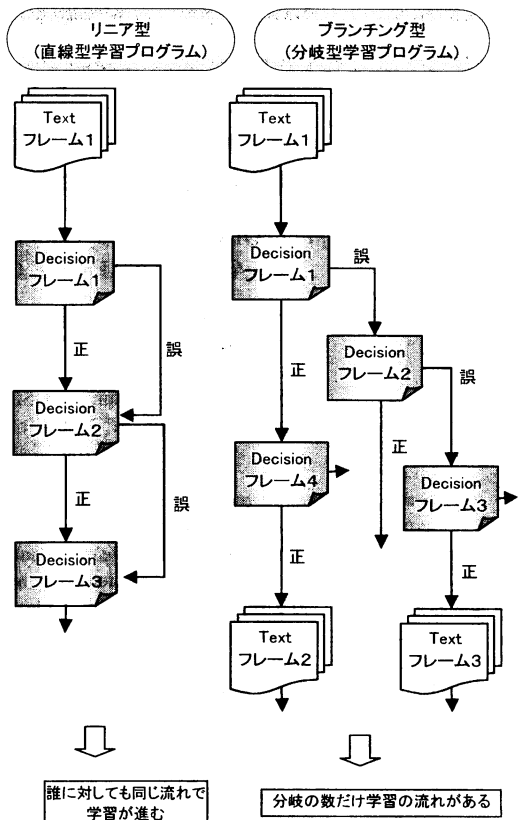


図1 教授ロジックのタイプ

3. 学習者モデル

学習者モデル[9]とは、学習者の理解状況をシステムが把握するための指標である。さらに詳しく言えば、学習者モデルとは、教材作成者が目標とする学習内容を、学習者が「どの部分を理解していないか」または「どの部分を誤って理解しているか」ということを明らかにするための指標である。このようにする理由は、「学習者が理解していない部分」が明らかにになれば、その部分を集中的に理解させるような教授ロジックを組み立てればよいからである。また、「学習者が誤って理解している部分」が明らかにになれば、その部分を矯正するような教授ロジックを組み立てればよいからである。このように、学習者モデルは教授ロジックを組み立てる上で基礎となる情報を取得するためのものである。これに対して、教材は教授ロジックを組み立てるための材料である。

ところで、学習者モデルにおける「学習者が理解した部分と理解していない部分」の把握と「学習者が誤って理解している部分」の把握では、教授ロジックを組み立てる上で、システムが把握すべき対象の粒度の大きさに違いがある。即ち、システムが把握すべき対象(=学習者に理解させるべき対象)の粒度が前者よりも後者のほうが遙かに細かいのである。例えば、学習者に理解させるべき対象が、前者では「ピタゴラスの定理」の公式であるのに対して、後者では「ピタゴラスの定理」の証明過程である。実は、このような粒度の違いが教材作成の容易さに大きな影響を与えている。即ち、前者では証明の結果である「ピタゴラスの定理」の公式さえ知っていれば、教材を作成できるのに対して、後者では「ピタゴラスの定理」の証明過程を理解していないと教材を作成できないのである。しかも、前者ではシステムがテキスト情報さえ扱えば教材を作成できるのに対して、後者ではグラフィック情報だけでなく、補助線を扱う機能も持たなければ教材を作成できないのである。ここでさらに、「三角形の2辺の和は他の1辺よりも大きいという定理」をも扱おうとした場合を考えると、教材作成の容易さに関して両者の間に大きな差があることはほぼ明白であろう。

学習者モデルは既に様々なモデルが提案されているが、前者との代表的なものとしてオーバレイモデルを挙げ、後者の代表としてバグモデルを挙げて、それぞれのモデルについて以下に説明する。

3.1 学習者モデルの例

(1) オーバレイモデル[5] [3] [9]

オーバレイモデルでは、出題の対象(=学習の対象)となるすべての問題に対して、表1のようなものを作成する。そして、例えば、問題番号 q の問題を正解した場合には表の q に対応する項目にフラグを立て、不正解の場合はフラグを立てない。このようにして、出題された問題の中で、理解できた(=正解した)問題のみにフラグを立てる。こうして、学習者がすべての問題に対して解答を入力し終わると、すべての学習対象に対して学習者が、どれだけ理解できたのかということが表現できる。つまり、オーバレイモデルは、学習者の理解を教師(=教材作成者)の知識の部分集合として捉えるモデルである。オーバレイモデルは学習者が習得した項目にフラグを立てる

だけでよいので、モデルの構築が非常に容易であり利用しやすいという特徴がある。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。表1では、問題番号2・4について理解できていないことがわかるが、何を理解していないかということはわからない。このように、オーバレイモデルは学習者モデルを単純にすることで、教材の作成は容易化できるが、学習者の理解状況を細かく把握するには適していない。

表1 オーバレイモデルの例

済/未	問題番号
✓	1
	2
✓	3
	4
...	...

(2) バグのモデル[9]

学習者の理解状況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。BUGGY[1]、DEBUGGY[2]、IDEBUGGY[2]などのシステムで用いられている BUGGY モデル[9]や模動モデル[12]などで代表されるバグのモデルは、学習者のバグ(=知識の不足、誤った理解、誤った理解を持つに至った過程、など)をモデル化することに重点が置かれている。バグのモデルは、オーバレイモデルでは表現できなかった学習者のバグが表現でき、学習者の理解状況を細かく把握することが可能となる。しかし、モデル構築からみると、バグのモデルは、学習者の理解状況を、事前に分析した学習者のバグの組み合わせで表現するため、準備が大変であり、教材作成者の負担がとてみ大くなるという問題がある。このため、バグのモデルは学習者の理解状況を細かく把握できるが、そのモデルに応じた教材を実現させるには非常に困難である。

3.2 本研究で採用する学習者モデル

前述したように、CAIシステムにオーバレイモデルを採用すると、他の学習者モデルよりも教材作成は容易になるが、学習者の理解状況の把握には適さないという問題がある。また、オーバレイモデル以外の学習者モデルを採用すると、学習者の理解状況の把握には適しているが教材作成は難しくなるという問題がある。そこで本研究では、学習者の知識の不足しか表現できなかった既存のオーバレイモデルを拡張して、学習者の不得意分野を把握できるようにした。これを“拡張オーバレイモデル”と呼ぶ。

3.3 拡張オーバレイモデルの具体例

本研究で拡張オーバレイモデルを使用する最大の目的は、学習者の不得意分野を把握することである。ところ

で、学習者に出題する問題は、必ず何かしらのカテゴリ(=出題分野)に属する。例えば「2進数 01011100 の 2 の補数を求めよ」という問題は「補数」というカテゴリに属する。もう少し細かく分類すると「2 の補数」というカテゴリに属するとも考えられる。さらに細かく分類することも出来るが、それは、教材作成者が学習者の理解状況を何処まで細かく把握したいかによる。このように、学習者に出題する問題には、必ず 1 つはカテゴリが存在する。本研究はそこに着目した。そこで、拡張オーバレイモデルは表2のような表を利用する。表2は、表1(=オーバレイモデル)を拡張し、各問題がそれぞれ属するカテゴリを表すキーワード(例えば、基数、補数、シフトなど)を記入する欄を設けられている。拡張の列が 3 つあるのはカテゴリ分けのレベルが 3 つあり、レベル I、II、III の順序で、より詳細なカテゴリ分けになっていることを示している。正誤の欄にフラグ(チェックマーク)が立っていれば、学習者がその問題には正解したことを示している。また、出題履歴の欄に設けている I、II、III の欄は分類カテゴリ(=キーワードのレベル) I、II、III に対応している。例えば、出題履歴の I の欄にフラグが立っている問題は、システムがレベル I の状態で学習者の不得意分野を把握する際に、使用したことを示している。

正誤	問題画面 番号	問題 番号	出題履歴			分類カテゴリ		
			I	II	III	I	II	III
✓	Java 00110	00110-01	✓			A,C	A1,C2	
		00110-02	✓			B	B1	
⋮		⋮	✓			⋮	⋮	
✓		00110-n1	✓			A,B,C	A1,B1,C1	
	Java 00120	00120-01		✓		B,C	B1,C1	
✓		00120-02		✓		A	A1	
⋮		⋮		✓		⋮	⋮	
		00120-n2		✓		B	B1	
	Java 00130	00130-01						
		00130-02						

表 2 拡張オーバレイモデルの全体像

図2は説明のために表2の一部分を取り出したものである。ここで、ある学習者が問題を5問解いて正誤判断した結果が図2になったとして、その学習者の不得意分野を把握する方法を説明する。この表を見れば、その学習者は「問題2」と「問題3」が不正解だったということが分かる。次に「問題2」と「問題3」が属するカテゴリ(ここではレベル I)を見ると、「問題2」は「基数」と「補数」に属し、「問題3」は「補数」に属することが分かる。そして、「問題2」と「問題3」は「補数」という共通のカテゴリに属することが分かる。その結果この表から、その学習者は「補数」という分野が不得意であると推測される。

このような拡張オーバレイモデルを使用すれば、システムは学習者の不得意分野を把握できるようになり、学習者の不得意分野に重点を置いた教材提示が可能となる。また、オーバレイモデルの特徴である教材作成の容

易さは維持されたままである。

拡張オーバレイモデル				
オーバレイモデル		拡張		
正誤	問題番号	カテゴリ(レベル I)		
✓	問題1	基数 シフト		
	問題2	基数	補数	
	問題3		補数	
✓	問題4	基数 シフト		
✓	問題5	シフト		

補数が苦手?

図2 拡張オーバレイモデルの具体例
(キーワードレベル I の部分のみ)

4. 本システムが採用する教授ロジック

本システムで採用する教授ロジックを図3のフローチャートで示す。

学習の流れ (フローチャート)

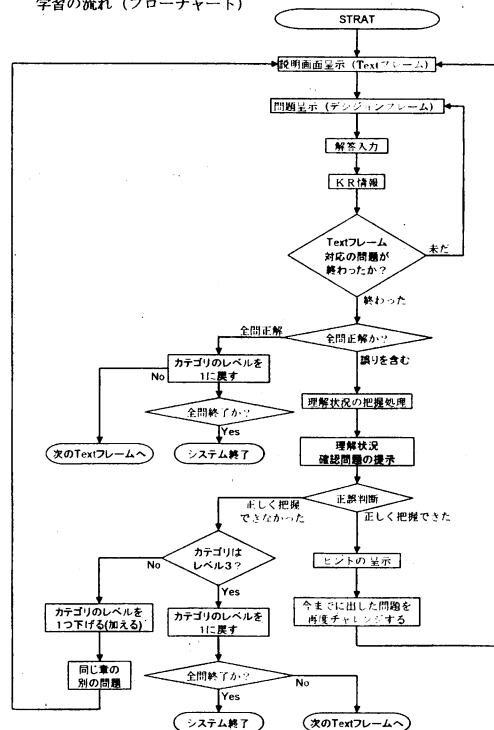


図3 本システムが採用する
教授ロジックの処理の流れ

まず、システムは1つの学習項目(説明画面の1画面分に相当する)の説明画面(Textフレームと呼ぶ)を呈示する。次に、システムは学習者に問題画面(デジジョンフレームと呼ぶ)を呈示し、解答させる。そして、システムは学習者の解答を正誤判断し、その結果(KR 情報)を学習者に呈示する。もし、学習者が問題を全て正解した場合は、システムはこのTextフレームには学習者の不得意分野が無いと判断し、学習者を次のTextフレームに進める。しかし、不正解があった場合には、システムはこのTextフレームには学習者の不得意分野があると判断し、学習者の不得意分野を把握する処理を行う。本システムでは、システムが学習者の不得意分野を把握する方法に帰納推論を用いている。以下に帰納推論を用いて、システムが学習者の不得意分野を把握する手順を示す。

前述したように、まず、システムは学習者が不正解した問題には、どのカテゴリに属しているかを調べる。次に、システムはそのカテゴリに重点を置いた問題画面を学習者に呈示し、解答させる。そして、システムは学習者の解答を正誤判断し、学習者が不正解した問題に属しているカテゴリを調べる。以上を繰り返すことで、システムは学習者が頻繁に不正解する問題の属するカテゴリの候補を取り出すことができる。しかし、システムは学習者の不得意なカテゴリを取り出すことができて、そのカテゴリが本当にその学習者の不得意分野であるかどうかは、この時点では判断できない。そこでシステムは、システムが不得意だと推測したカテゴリが本当に正しいのか、確認するための問題画面を学習者に呈示し、解答させる。確認するための問題画面には、システムが推測した学習者の不得意分野に重点が置かれた問題画面になっているが、不得意分野と推測されていないカテゴリに属する問題もその中には含まれている。これは、不得意分野の範囲を拡大して推定する誤りを防ぐためである。そして、学習者の解答結果が不得意だと推測したカテゴリに属している問題が全て不正解になると、システムが推測した結果は正しいことになり、システムはそのカテゴリが学習者の不得意分野だと把握できたことになる。このように本システムでは学習者の不得意分野を把握する方法に帰納推論を用いている。

しかし、システムが不得意だと推測したカテゴリが本当に正しいのか、確認するための問題画面で、学習者の不得意分野に属する問題が全て不正解になるとは限らない。そこで本研究では、その場合、学習者の不得意分野の推測が誤りであったとは考えず、推測したカテゴリでは分類した範囲が大きすぎたために不得意ではない分野も含まれてしまい、不正解にはならないものが含まれてしまったと考える。そこで本システムは、より詳細な分類になっているカテゴリレベルⅡへ移行する。レベルⅡでもシステムは、前述したレベルⅠと同様の作業(説明画面の呈示から学習者の不得意分野を把握するまで)を行う。また、レベルⅡでもシステムが学習者の不得意分野を正しく把握できなければ、システムはさらに詳細な分類になっているカテゴリレベルⅢへ移行する。しかし、レベルⅢでもシステムが学習者の不得意分野を正しく把握できない場合は、仕方なく学習者を次のTextフ

ームに進める。

また、本システムでは、システムが学習者の不得意分野を把握した時点で、その不得意分野に重点を置いた説明画面(ヒント)を学習者に呈示する。そして、学習者はその説明画面をみた後、今までに出題した問題を再度チャレンジしてもらうことになる。

5. AFO-CAI と rule-based CAI

商用として実際に利用されているWBT型CAIシステムでは、システムの構築方法に、主に AFO-CAI (Ad-hoc Frame Oriented-CAI)[13]を用いている。AFO-CAI は、予め予想される反応を細かく分析し、配列された学習プログラムに従って行われる。この手法では教材と教授ロジックが一体に(物理的に一つのもの)となっており、教材画面の提示順序を変更するには、教材提示を司るメインプログラムの教授ロジック(教材提示アルゴリズム)を変更しなければならない。分岐型の CAI システムでは、教材画面の提示順序を変更する作業が頻繁に発生する。しかし、教材画面の提示順序を変更するたびに教材提示プログラムを書き換えるのは効率が悪い。また、教授ロジックを作成する教材作成者は、教材提示プログラムを書き換えることができるだけの、プログラミング能力が必要となる。これは、教材作成者にとって、とても負担がかかることである。

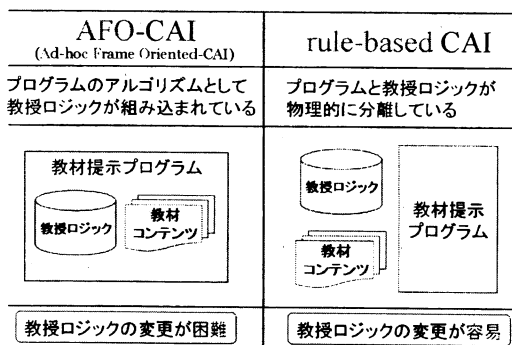


図4 AFO-CAI と rule-based CAI の比較

一方、rule-based CAI は、教授ロジックが if-then ルールの集合になっており、教材提示プログラムが教授ロジックや教材コンテンツとそれぞれ物理的に独立している。そのため、教材提示プログラムを変更することなく、if-then ルールを変更するだけで教授ロジックを変更できる。従って、rule-based CAI では、教材作成者は試行運用しながら教授ロジックを改良していくことができるので、より良い分岐型教材の作成が可能となる。よって、本システムが採用するCAIシステムの構築方法はrule-based CAI である。

6. 作成した教材の例

ここでは、第4章で述べた本システムが採用する教授ロジックの流れを具体的な教材例と共に示す。例として、プログラミング言語Javaの文法教材について説明する。その中の一例として、「演算子」の学習項目を採り上げる。

演算子には、様々な種類があり、また、使用するにあたり個々のルールを知っておかねばならない。まず、学習者に対して「演算子」というキーワードに含まれる学習項目をテキストフレームに提示する。(6.1テキストフレーム参照) ここで、フレームとは、ディスプレイ1画面に表示できるようにまとめられた説明文や疑問文のことである。以降の説明では、教材画面をフレームという単位で説明する。続いて、演算子というキーワードに含まれる問題を図6に示す。演算子には、代入・単項・算術・論理・ビット・条件・関係・インクリメント/デクリメント演算子が含まれる。これらのキーワード全てを含む説明画面(Text フレーム)を学習者に提示する。後に、学習者は説明画面で学習した内容についての問題(デジジョンフレーム)へと移る。デジジョンフレームでは、先ほどの演算子の学習項目中に存在する様々な問題に解答していく。解答した結果、CAIシステムはKR(Knowledge of result)を学習者に提示する。テキストフレーム対応のデジジョンフレームはいくつか存在し、このフレームごとにKR情報を提示することになる。そして、テキストフレーム対応のデジジョンフレームを学習者が完了した場合は、解答状況を処理し全問正解の時は次のテキストフレームへと移る。もし、ここで、学習者が解答した問題に誤りがあった場合は、理解できていないとシステムが判断し、理解状況の把握の処理へ移る。情報で弱い項目へと学習画面が移っていく。ここでは、学習がどの程度達成されたかをシステムが機能推論することによって判断する。理解状況の把握では、先ほど解答した問題が、たまたまできなかったのか、或いはその学習者に対しては、教材の粒度と難易度が高かったのかを確認する処理が必要である。それを行った上で、正誤判断によりシステムが正しく理解状況を把握できた場合にはヒントの提示によって先ほど解答した問題が容易に解ける。ここで、重要なのは、正しく判断できなかった場合にどのような教材を学習者に提示するかである。本研究では、始めに学習した教材の理解が得られていないとシステムが判断した場合、教材に問題があると捉えて学習者に対してはより粒度の小さい教材を提示する。以下では、作成した教材を用いて学習の流れをフローチャートの図と照らし合わせながら説明する。

6.1 テキストフレーム

テキストフレームには、プログラミング言語 Java の文法に関する項目より、「識別子」についての一般的な説明画面が掲載されている(図5参照)。この「識別子」の項目を学習するときには、どの学習者もこの一般的な説明画面をみて学習を行う。しかし、どの学習者も理解度に差があるため、同じように学習して理解できるとは限らない。そこで、最初のテキストフレーム中にある説明項目をもっと詳細にして教材の粒度を小さくする。つまり、最初のテキストフレームで十分理解できる学習者もいるが、実際にデジジョンフレームに移行したときに思うような正当率が得られない場合に、欠落している知識を強化するために次のテキストフレーム(教材粒度の詳細版)を用意しておく。そうすることによって、理解度にばらつきのある学習者にこのシステムが対応できる。

テキストフレーム (Java00110)

演算子
変数や定数を用いて各種演算を行う記号を演算子といいます。このとき、演算に用いられるデータのことをオペランドと呼びます。

以下の表は、java言語で利用可能な演算子です。処理優先度の高いものが上に書かれていますが、一般的な数学の法則に沿うものです。

図5 テキストフレームの例

6.2 デジジョンフレーム

図5のデジジョンフレームでは、3つの問題例が掲載されているが、いずれも「識別子」というキーワードに属する問題である。このデジジョンフレームには、拡張オペレイモデルの表にある分類カテゴリがいくつも存在する。それは、代入・単項・算術・論理・ビット・条件・関係・インクリメント/デクリメント演算子それぞれに関する知識の問いかけの問題である。また、一つの問題で複数の知識が必要な問題も存在する。学習者がこのデジジョンフレームに解答をしていく過程を説明する。このデジジョンフレームは、その直前に提示されたテキストフレームの知識を学習してから取り組むフレームである。「識別子」に関する一般的な説明を学習してからこのデジジョンフレームに取り組む。しかしながら、一般的な識別子に関する解説画面の説明で学習を行っても全ての学習者がここで、提示されたデジジョンフレームの問題画面に全て正答できるとは限らない。したがって、間違った問題(学習者の弱い学習項目)について教材画面が分岐していくのが望ましい。以下、フローチャートを用いて学習の流れを説明する。

デジジョンフレーム (Java00110)

問題00110-01
次の①～④の説明に当てはまる演算子を(a)-(i)からすべて選びなさい。
①代入演算子
②算術演算子
③単項演算子
④インクリメント演算子
選択肢: (a)+ (b)++ (c)= (d)== (e)> (f)>> (g)& (h)&& (i)?

問題00110-02
次の式の計算をしなさい。I×Yの値はいくらか?
int I = 1;
int X = ++I;
int Y = ++X;

問題00110-03
次のコードを実行した後のYの値はどれか?
int x = 5;
int y = 4;
y = x++;

選択肢: (a)4 (b)5 (c)6 (d)7

図6 デジジョンフレームの例

6.3 作成した教材を用いた学習の流れ

理解度に応じた学習項目の分岐を実現しているのが以下に示すフローチャートの分岐部分である(図7, 8, 10参照)。先ほどから述べてきた理解度に応じた学習項目の分岐するポイントをそれぞれ注目して説明する。

(1) 学習者の理解度推論

学習者の理解度を推論する部分は、一度解答して間違えた問題と似た(同じレベルの違う問題)を再度提示してその問題が解けるかどうかで判断する。また、推論が正しい場合は、その問題に対してヒントを提示することに

よって再度同一問題にチャレンジした場合には正答率が高くなると予想できる。

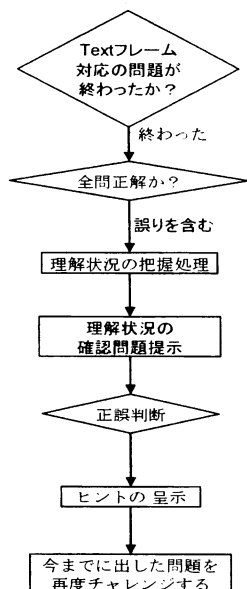


図7 学習者の理解度推論

(2) 新しい学習項目への分岐

呈示されたテキストフレームを理解し、次に呈示されたデジジョンフレームにおいて正答率が高くテキストフレーム対応のデジジョンフレームの問題に解答が終わったときは、新しい項目のテキストフレームに移行する。ここでは、テキストフレーム対応の問題解答が全ておわり、全問正解であった場合は、新しい学習項目に移ることにする。

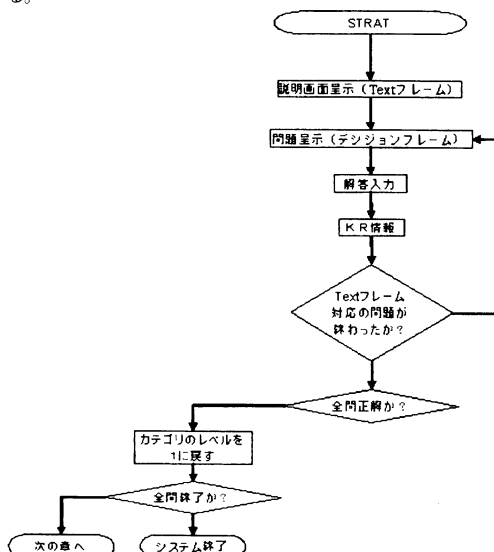


図8 新しい学習項目への分岐

(3) 粒度の細かいデジジョンフレーム

先ほどの場合とは違って、理解が不十分であり、呈示されたデジジョンフレームにおいては正答率が低かった場合、システムは図9のような教材の粒度が細かい学習項目へと分岐する。この呈示方法は、学習者が理解に苦しみやすい学習内容であったり、苦手としている項目を克服するのにに対して効果的といえる。ここでは、デジジョンフレームの内容が、先ほどはさまざまな演算子についての問題であったが、学習者の苦手としている「インクリメント・デクリメント演算子」について着目してシステムが問題呈示を行っている。

デジジョンフレーム (Java00120)

```

問題00120-01
次の式の計算をしない。bo,i,jの値はいくつか?
int i = 1;
int j = 1;
boolean bo = ++i > j++;
問題00120-02
次の式の計算をしない。d,i,jの値はいくらか?
int i = 1;
double d = 5;
i = (int)d++;
j = (int)d - d;
問題00120-03
次の式の計算をしない。bo,iの値はいくつになりますか?
int i = 1;
boolean bo = ++i < i++;
  
```

図9 粒度の細かいデジジョンフレーム

(4) 理解度に応じた学習項目の分岐

最後に、学習者の理解度に応じて適した教材を呈示するフローチャート部分の説明をする。教材の粒度を細かくすることによって、学習者は効率の良い学習が行うことができると言える。ここでは、粒度の細かな教材に分岐するだけでなく、レベルを設けている。

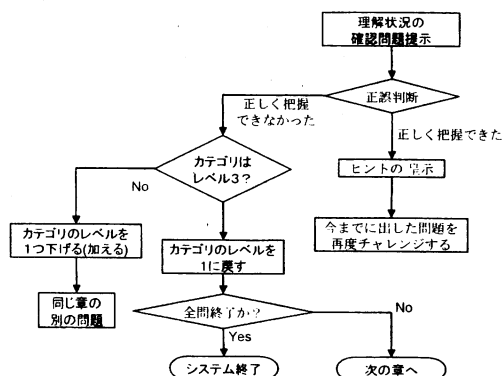


図10 理解度に応じた学習項目の分岐

7. おわりに

本論文では、教授ロジックには直線型と分岐型があり、誰に対しても同じストーリーで教材を提示する(即ち、提示する教材画面がその提示順序を含めて同じになる)直線型教授ロジックよりも、提示された問題に正解するか

否かによって、次ぎに提示される教材画面が動的に変化する分岐型教授ロジックのほうが学習効率が低いことを指摘した。にも拘わらず、市場に出回っているCAIシステムのほとんどは教授ロジックを採用している。その理由は、教授ロジックに合わせて教材を具体的に組み立てる作業が困難だからである。この問題を解決するために、rule-based システムとしてCAIシステムを構築することを提案した。次ぎに、分岐型教授ロジックの考えをさらに押し進めて、システムが学習者の理解状況(何処が学習者の不得意分野であるか)を把握することにより、学習者が苦手としている分野へ向って教材画面がとどンドン分岐して行くような教授ロジックのシステムを構築することを提案した。そのようなCAIシステムを構築するために、オーバーレイモデルを拡張した学習者モデル(拡張オーバーレイモデルと呼ぶ)を提案するとともに、それに基づく教授ロジックの構築法(フローチャート)を示した。そして、教材の具体例を示すことにより、拡張オーバーレイモデルとそれに基づく教授ロジックの有効性を示した。

[参考文献]

- [1] S. J. Brown and R. R. Burton, "Diagnostic Models for Procedural Bugs in Basic Mathematical Skills," *Cognitive Science*, Vol. 2, No.2, pp.155-191, 1978.
- [2] R. R. Burton, "Diagnosing Bugs in a Simple Procedural Skill," H. D. Sleeman and J. S. Brown, eds., *Intelligent Tutoring Systems*, Academic Press, London, pp.157-183, 1982.
- [3] B. Carr and I. Goldstein, *Overlays: A Theory of Modeling for Computer Aided Instruction*, "MIT AI Memo 406, 1977.
- [4] Norman A Crowder, "On the differences between linear and Intrinsic programming," *Phi Delta Kappan*, Vol. 44, pp. 250-254, 1963.
- [5] I. Goldstein, "The Computers as Coach: An Athletic Paradigm for Intellectual Education," MIT AI Memo 389, 1977.
- [6] 堀内敏夫編, "プログラム学習とTM," 教育工学講座4, 大日本図書, 1972.
- [7] 平山加菜、小野寺直樹、山本洋介、橋浦弘明、古宮誠一, "ブランディング型教授ロジック実現のためのISO型CAIシステム," 第66回情報処理学会全国大会講演論文集, vol.5, 5C-04, Mar. 2004.
- [8] 北尾倫彦, "学習の心理<TP 叢書>," ミネルヴァ書房, 1978.
- [9] 溝口理一郎、角所収, "知的CAIにおける学習者モデル," 情報処理, Vol. 29, No. 1, pp.1275-1282, 1988.
- [10] B. F. Skinner, "The Science of Learning and the Art of Teaching," *Harvard Educational Review*, Vol.24, pp. 86-97, 1954.
- [11] B. F. Skinner, "Teaching Machines," *Science*, Vol. 128, pp. 969-977, 1958.
- [12] D. H. Sleeman, "Inferring (Mal) Rules from Pupiles' Protocols," *Proc. Of European Conference*

on Artificial Intelligence, Orsay, France, pp. 160-164, 1982.

- [13] 山下英男監修, 共立総合コンピュータ辞典, 日本ユニバック総合研究所編, 共立出版, pp.891-908, 1990.