

eラーニングにおける学習シーケンシングの妥当性検証

小西 裕[†] 延原 哲也^{††} 劉 渤江^{†††} 国島 丈生[†] 横田 一正[†]

[†] 岡山県立大学大学院 〒 719-1117 岡山県総社市窪木 111

^{††} 岡山理科大学 〒 700-0005 岡山市理大町 1-1

^{††} (株) ベネッセコーポレーション 〒 700-8686 岡山市南方 3-7-17

E-mail: [†]{konishi,kunishi,yokota}@c.oka-pu.ac.jp, ^{††}tetsuya_nobuhara@mail.benesse.co.jp ,
^{†††}liu@mis.ous.ac.jp

あらまし eラーニングでは、学習者にどのような教材を提示するかの学習シーケンシングが重要な役割を果たしている。学習シーケンシングは一般的に学習者の学習レベルやテスト結果に基づく条件付き有向グラフとして表現される。教材作成者が学習シーケンシングの設計を行った際、誤りにより矛盾を含んだグラフになることが考えられる。本稿では、ECA ルールで記述された学習シーケンシングから条件付き有向グラフを抽出し、構造的な矛盾と意味的な矛盾の分類から、それらの矛盾の検出方法を提案する。

キーワード eラーニング, ECA ルール, シーケンシング, 妥当性検証

Validation Checking of Learning Sequencing in e-Learning

Yutaka KONISHI[†], Tetsuya NOBUHARA^{††}, Bojiang LIU^{†††}, Takeo KUNISHIMA[†], and
Kazumasa YOKOTA[†]

[†] Okayama Prefectural University 111 Kuboki, Soja, Okayama, 719-1117 Japan

^{††} Okayama University of Science 1-1 Ridai-cho, Okayama, 700-0005 Japan

^{††} Benesse Corporation 3-7-17 Minamigata, Okayama 700-8686 Japan

E-mail: [†]{konishi,kunishi,yokota}@c.oka-pu.ac.jp, ^{††}tetsuya_nobuhara@mail.benesse.co.jp ,
^{†††}liu@mis.ous.ac.jp

Abstract In e-learning, sequencing plays an important role in what kind of teaching materials should be presented for learners. Sequencing is generally represented in the form of a conditional directed graph based on learners' understanding levels or test results. However, there might be errors or contradictions in sequencing when teachers design and make teaching materials. In this paper, we propose to extract a conditional directed graph from sequencing represented in ECA rules and validate both syntactic and semantic errors in the graph.

Key words e-Learning, ECA Rule, Sequencing, Validation Checking

1. はじめに

近年、社会全体へのブロードバンドの普及を背景にして、企業内教育や高等教育を中心に eラーニングを導入する事例が増加している。学習者がネットワークを利用して自主学習を行う Web ベースの eラーニングにおいては標準規格である SCORM(Sharable Content Object Reference Model) [1] が ADL(Advanced Distributed Learning Initiative) により提唱されている。

SCORM では「テストが悪ければ復習を行う」といったような学習順序の制御(シーケンシング)を行う方法が定められているが、実際は単純化された成績による最低限の制御しかでき

ないため学習者の成績やつまづき箇所に対応するための十分な機能を備えていない。

我々はこれまで SCORM における問題の解決策として、解答の正誤ボタンと ECA ルールを利用した形成的評価のためのシーケンシングの研究 [5] を行ってきた。更にその内容に加えて、より詳細な学習者状態の診断を行い、診断結果に合わせて様々な教材を動的に提示する、適応型学習機能を備えるシステムモデルを提案した。[6]

適応型 eラーニングシステムにおいて、教材の提示順の制御は学習シーケンシングにより行われる。学習シーケンシングは教材作成者により設計されるが、その際に誤りによりシステムの動作に支障をきたすようなシーケンシングを作成してしまう

事が考えられる。あるいは動作上問題は無くとも、作成者の意図を反映していない場合もありうる。

そこで本稿では、上述した作成者による設計ミスを検出するため、学習シーケンシングの妥当性の検証を行う手法を提案する。まず、学習シーケンシングから教材の提示順の順序関係を抽出し、教材の提示順をテスト結果や学習者レベルに基づく条件付き有向グラフとしてモデル化する。またシステムの動作に支障をきたす場合をシーケンシングの構造的な矛盾、教材作成者の意図を反映していない場合を意味的な矛盾として定義する。モデル化したグラフと矛盾の分類から、学習シーケンシングにおける矛盾の検出方法について述べる。

2章ではECAルールにより記述される学習シーケンシングについて述べる。3章では学習シーケンシングにおける構造的な矛盾と意味的な矛盾の定義と妥当性検証の流れ、及び矛盾の検出方法について述べる。4章では実装した学習シーケンシングの妥当性検証器とその評価を示し、5章で結論と今後の課題を述べる。

2. ECAルールによる学習シーケンシング

2.1 ECAルール

ECAルールとはアクティブデータベースの分野で用いられていたルールの記述方式で、Event、Condition、Actionの3つを一組にして記述する。以下で示すEventは教材提示画面における学習者の操作によるものと、後述する学習者情報の更新によるものの2種類がある。また、ConditionとしてEvent発生時に決まる値と学習者情報を用いることができる。Actionとしては教材の提示や学習の終了、学習を支援するためのメッセージの送信などがある。

Event

Eventは「次の教材モジュールを要求」、「任意の教材モジュールの要求」、「難易度変更の要求」、「学習者情報の更新」の4種類である。

Condition

Conditionは「学習者情報」、「Event引数」の2種類である。

Action

Actionは「教材モジュールの提示」、「全体を再学習」、「学習の終了」、「学習者レベルの変更」、「メッセージの送信」、「補助教材の提示」の6種類である。

これらのルールを複数組み合わせることにより、SCORM2004 [1]で採択されているIMS Simple Sequencing 1.0 [2]相当のシーケンシング制御を実現することも可能である。

2.2 学習者情報

学習者情報として教材モジュール閲覧履歴、テスト結果、学習者レベルの3つがある。これらの情報に基づいて実行すべきActionを決定する。

- 教材モジュール閲覧履歴

ある学習シーケンシングの中で過去に提示されてきた教材モジュール、現在提示している教材モジュールの識別子の情報を持つ。さらには各教材モジュールを提示した回数や提示した時間(秒)の情報を持つ。

- テスト結果

これまでに行ってきたテストの結果である。各テストごとに解答の正誤パターンと解答に費やした時間、そして結果として合格・不合格の情報を持つ。

- 学習者レベル

上級者、中級者といった学習者のレベルを表す。学習者レベルを用いることにより教材モジュール内の提示順序を切り替えることができる。教材モジュールの具体的な記述については文献 [7]で述べる。

2.3 シーケンシングのモデル

学習シーケンシングは教材の提示順序や学習者へのメッセージの送信といった学習の流れを制御する。また、条件付き有向グラフとしてモデル化され、学習の開始地点となる頂点と終了地点となる頂点を一つずつ持つ。条件付き有向グラフとは、頂点間を結ぶ辺に移動の条件となるラベルが付加されている有向グラフである。モデル化したシーケンシングの例を図1に示す。グラフの各頂点のラベルは、学習者に対して一度に提示する単位の、教材の識別子を表している。この提示単位を以後「教材モジュール」と呼ぶ。またグラフの各辺はEvent検知によるAction「教材モジュールの提示」の動作を表している。辺につけられる「テストBに合格」等のラベルはルールの発火条件であるConditionを表す。Action「教材モジュールの提示」は教材モジュールの提示の指示を出した後、教材モジュール閲覧履歴にそのモジュールの提示を行ったという情報を追加する。図1の場合では、頂点Aの次に頂点Bの内容が提示され、Bにおけるテストの解答結果より次に提示する頂点分岐する。合格すれば終了地点Dに到達し、不合格ならばC、Eを経てBが再び提示される。また、これらの頂点の提示内容は教材モジュール群を参照することで決定する。

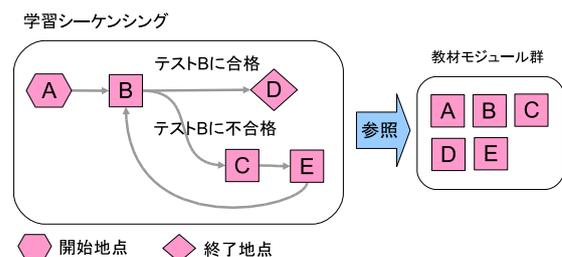


図1 学習シーケンシングと教材モジュール

2.4 学習シーケンシングの記述例

ここではECAルールによる学習シーケンシングの記述の例を示す。教材作成者は図1を設計するため以下に示すような形式でECAルールを記述する。

Event 次の教材モジュールを要求

Condition 現在提示している教材モジュール=A

Action 教材モジュールBを提示

Event 次の教材モジュールを要求

Condition 現在提示している教材モジュール=Bかつ

テストBに合格

Action 教材モジュール D を提示
 Event 次の教材モジュールを要求
 Condition 現在提示している教材モジュール=B かつ
 テスト B が不合格
 Action 教材モジュール C を提示

C から E を経て B を再提示するためのルールは省略する。2 番目、3 番目のルールでは学習者のテスト B の結果によって発火するルールが変わるため、提示順序が分岐しているシーケンシングを表現することができる。

3. 学習シーケンシングの妥当性検証

学習シーケンシングの妥当性は次に示す要件を満たすことで保障される。途中で途切れることなく学習を行えること、学習の目標地点に必ず到達することができること、教材作成者の意図通りの順序で教材モジュールが提示されること、作成者の用意した教材モジュールが全て利用されていることの以上 4 点である。

妥当性を検証するためのアプローチを述べる。検証は条件付き有向グラフとしてモデル化した学習シーケンシングと、そこで参照する教材モジュールとの関連性を対象に行う。また、学習シーケンシングを条件付き有向グラフにモデル化する。それによって一般的なグラフ探索アルゴリズムを妥当性検証に利用することができる。更にシステムの動作に支障をきたす場合、あるいは教材の提示順として妥当でない場合のグラフの構造や頂点間の移動の条件を矛盾として定義することができる。また、モデル化は教材の提示順を教材作成者に視覚化して提示することにつながり、グラフの構造などの情報からは決して推測できない教材作成者の錯誤の発見をサポートすることができる。

Paul らによる研究 [3] では、コンテンツの提示順序を adaptation rule により制御する方式を提案している。ルールが無制限ループに陥ってしまうのを防止するため、ルールの上位概念であるグラフを設計し、その構造からルールの自動生成を行う。しかし、グラフ自体の妥当性検証の方法は示されておらず、その点で本稿で提案する内容とは異なっている。

3.1 構造的な矛盾の定義

シーケンシンググラフの矛盾

図 2(a) では頂点 C, E 間で遷移がループしているため終了地点に到達できない構造となっている。また、図 2(b) ではグラフが分割されているため同様に到達不可能である。

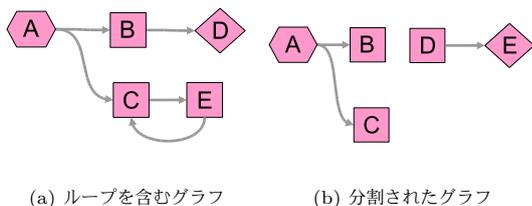


図 2 シーケンシンググラフの矛盾の例

分岐条件の不足/重複の矛盾

図 3 で条件として与えられている test1, [1 0] は test1 のテストの解答結果の正誤パターンが“正解”, “不正解”であることを表している。テストの解答結果による分岐では点数を利用することも考えられるが、本稿では検証手法の簡単化のため正誤パターンのみを扱っている。図 3(a) では条件のパターンが“0 0”, “0 1”のときの遷移が定義されていない。また、図 3(b) ではパターンが“1 1”のときにどちらに遷移すべきか分からない。なお、ここで対象とする矛盾は、条件の取り得る値が事前に全て分かっているという前提のもと定義している。

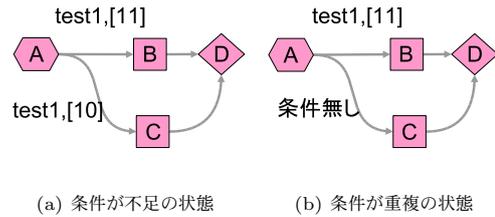


図 3 分岐条件の不足/重複の矛盾の例

教材モジュールへの参照の矛盾

図 4 の例では実際には存在しない教材モジュールの識別子を参照しているため該当の教材モジュールを提示することができない。

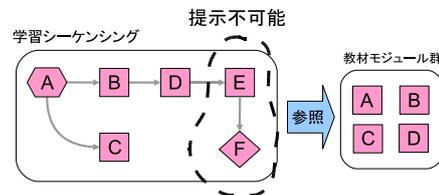


図 4 教材モジュールへの参照の矛盾の例

3.2 意味的な矛盾の定義

経路上での移動条件の矛盾

図 5 の例では A から B への条件と、B から D への条件は論理的に同時に成立することは有り得ない。

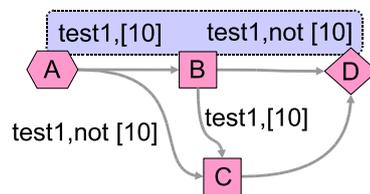


図 5 経路上での移動条件の矛盾の例

テスト結果による再学習の矛盾

一般的にはテストに解答した後、得点が基準に達していなければその単元の再学習を行うという順序の学習が考えられる。図 6 の例では、C において test1 の結果が全問正解のときに以前に提示した教材モジュールを提示し、それ以外なら先に進むよう定義されているため、学習シーケンシングとして見たときに妥当ではないと判断できる。しかし、この検証の有効性は教

材作成者の作成する教材の目的に依存するので、検出の際には警告メッセージを出すにとどめる。

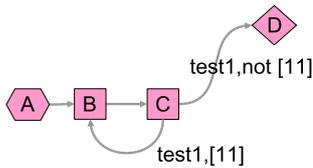


図 6 テスト結果による再学習の矛盾の例

3.3 構造的な矛盾の検証方法

3.3.1 シーケンシンググラフの検証

シーケンシンググラフの検証はグラフの構造から終了地点に到達不可能な頂点を教材作成者に提示することによって実現する。また、開始地点から到達することのできない頂点と分割グラフの検出も行う。条件付き有向グラフは隣接点行列で表現される。表 1 に概念を示す。隣接点行列とは二つの頂点間の辺の有無を行列で表したもので、ここでは行側の頂点が遷移元、列側の頂点が遷移先を表している。

表 1 隣接点行列

＼	頂点 1	頂点 2	...	頂点 N
頂点 1	0	0	...	1
頂点 2	1	0	...	0
⋮	⋮	⋮	⋮	⋮
頂点 N	0	1	...	0

検証の手順を述べる。隣接点行列に Warshall 法 [4] を適用し、グラフの推移閉包を求める。推移閉包とは、ある頂点から頂点へ到達可能な経路が存在するとき、その頂点間を直接結ぶ辺を追加し、新たにグラフを構成したものである。

こうして求めた有向グラフの推移閉包より、全ての頂点が終了地点に到達可能であるかを検証する。図 2(a) のグラフから Warshall 法の適用により導出した推移閉包を表 2(a) に示す。

表 2 推移閉包

＼	A	B	C	D	E
A	0	1	1	1	1
B	0	0	0	1	0
C	0	0	1	0	1
D	0	0	0	0	0
E	0	0	1	0	1

(a) ループを含むグラフの推移閉包

＼	A	B	C	D	E
A	1	1	1	0	0
B	1	1	1	0	0
C	1	1	1	0	0
D	0	0	0	1	1
E	0	0	0	1	1

(b) 無向分割グラフの推移閉包

表 2(a) より終了地点 D に到達できない頂点は C と E であることが分かる。また、開始地点 A から全ての頂点に到達できることが分かる。

次に、分割グラフの検証方法について述べる。与えられた有向グラフを無向グラフへ変換し、同様に Warshall 法を適用して推移閉包を導出する。図 2(b) のグラフに対してこの方法を

用いると推移閉包は表 2(b) のようになる。これより A, B, C のグラフと D, E のグラフが分割されていることが分かる。

3.3.2 分岐条件の不足/重複の検証

種々の分岐条件を統一的、かつ過不足なく設計されているか検証するため、表 3 に示したコストパタンテーブルを用いる。コストパタンテーブルとは、ある分岐の中で使われている条件の組合せのパタン全てを表したものである。条件の真偽を“1”、“0”で表し、n 個の条件に対するパタンの数は 2^n 個である。コストパタンテーブルを生成した上で各遷移の条件をパタンの形に変換し、マッチングを行うことで条件の不足/重複の検証を実現する。

表 3 コストパタンテーブル

桁	n	...	2	1
条件	条件 n	...	条件 2	条件 1
コスト	0	...	0	0
パタン	0	...	0	1
	⋮	⋮	⋮	⋮
	1	...	1	1

パタンマッチングの手法について述べる。遷移の条件のパタンは“1”、“0”、“*”の3つの値で表現される。*はワイルドカードを意味し、1と0の両方に対応する。この条件のパタンとコストパタンテーブルの各パタンとで、次に示す排他的論理和を拡張した演算規則に従ってビットごとの演算を行う。演算結果が全て0になった場合、パタンは“マッチした”とする。

$$\begin{aligned}
 1 \oplus 1 &\rightarrow 0 & 0 \oplus 0 &\rightarrow 0 \\
 1 \oplus 0 &\rightarrow 1 & 0 \oplus 1 &\rightarrow 1 \\
 * \oplus 1 &\rightarrow 0 & * \oplus 0 &\rightarrow 0
 \end{aligned}$$

次に、分岐条件の矛盾の判定方法を述べる。テーブル内のあるパタンが複数の条件でマッチした場合、そのパタンの条件による遷移が重複して定義されていると判断できる。また、テーブル内のあるパタンがどの条件でもマッチしなかった場合、そのパタンの条件による遷移が定義されていないと判断できる。テーブル内の全てのパタンが単一の条件とマッチした場合に分岐条件が過不足なく定義されていると判断することができる。

図 3(b) の A の分岐でのコストパタンテーブルを表 4 に示す。A から B への遷移の条件は“1 1”なのでテーブルの“1 1”とだけマッチする。C への遷移の条件は定義されていないためパタンは“* *”となる。演算規則よりテーブルの全てのパタンとマッチするため“1 1”の条件において重複の状態であると判断できる。

表 4 条件数 2 個のコストパタンテーブル

桁	2	1
条件	test1	test1
コスト	0	0
パタン	0	1
	1	0
	1	1

3.3.3 教材モジュールへの参照の矛盾の検証

ここでは教材モジュールへの参照の矛盾の検証方法を述べる。学習シーケンシングの順序関係を抽出しグラフ化を行う際、教材モジュールの実際の識別子をリストとして取得する。次にグラフの頂点のラベルのリストと、実際の識別子のリストを比較する。実際の識別子のリストに存在しなかったラベルの頂点に矛盾であるという情報を付加して、教材作成者に提示することによって実現する。

3.4 意味的な矛盾の検証方法

意味的な矛盾の検証は、構造的な妥当性が保障されているという前提のもとで行うものとする。また、3.2節で定義した矛盾のうち“テスト結果による再学習の矛盾”についての検証方法を述べる。

3.4.1 テスト結果による再学習の矛盾の検証

テスト結果の再学習の矛盾の検証方法について述べる。この検証の対象となる条件は、テスト結果の解答の正誤パタンのみである。

テストの解答結果を条件として用いる分岐について、3.3.2で述べた検証方法を応用する。最初に、グラフの中で過去に提示した頂点へ移動する辺を、深さ優先探索を用いて導出する。ここで導出された辺を“再学習のための遷移”とする。次に、分岐条件の正誤パタンからコストパタンテーブルを生成する。分岐の中で、再学習のための遷移の条件のパタンと、パタンテーブルとのマッチングを行う。マッチしたパタンの中で1が最も多いパタンと、マッチしなかったパタンを順次比較していく。前者より1の数が少ないパタンが見つかったならば、再学習のための遷移の条件に正答数の多いパタンがつけられていると判断できる。

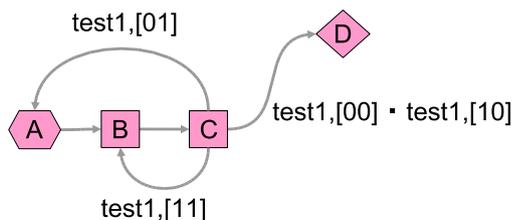


図7 テスト結果による再学習の矛盾の検証例

図7の例では再学習のための遷移の条件のパタンは“0 1”と“1 1”であり、残りのパタンと比較すると1の数が多いため分かる。このことから再学習を行う条件のパタンが学習シーケンシングとして妥当ではない可能性があるという判断ができる。

4. 妥当性検証の実現

4.1 学習シーケンシングの Prolog による記述

学習シーケンシングは前述したように ECA ルールで定義するが、具体的には Prolog で記述する。Prolog は“事実”と“規則”で表現される節の集合からなる論理型プログラミング言語である。定義された事実と規則から、推論によって解を導く事ができるという特徴がある。このことから、“Event かつ Condition ならば Action である”といったルールを記述す

る上で、他の手続き型言語などの制御構文、if-then-else 文や switch-case 文に比べて直観的に記述しやすい言語であると言える。また、妥当性検証を行う上でも容易に教材提示の順序関係を抽出でき、扱いやすい記述形式である。

教材作成者は図8に示すように学習シーケンシングを記述する。Event として、次の教材モジュールの要求が発生したときに発火するルールを、述語 next(From, To, LI, Action) で記述する。それぞれの引数の意味は次の通りである。From は現在提示している教材モジュール、To は次に提示する教材モジュール、LI は学習者情報、Action では教材モジュールの提示以外の Action を指定する。また、学習者情報の参照は、引数 LI により渡された学習者情報を述語を用いて参照する。図8中の:-testPassed(LI,"test1","passed") は test1 の結果が合格であったときに真となる条件である。また、不合格は“failed”で表される。

```
% 次の教材モジュールを要求
% next(From, To, LI, Action):-Condition

next("start","id1",_ , []).
next("id1","id2",LI, []):-
    testPassed(LI,"test1","passed").
next("id1","id4",LI, []):-
    testPassed(LI,"test1","failed").
next("id2","id3",_ , []).
next("id3","end",_ , []).
next("id4","id3",_ , []).
```

図8 学習シーケンシングの記述例

図8の学習シーケンシングでは id1 の教材モジュールから始まり、そこでテストの結果により id2 と id4 に分岐し、id3 で学習が終了することを表している。

4.2 妥当性検証器の実装

学習シーケンシングの妥当性を検証する妥当性検証器を Java で実装した。3.章で定義した矛盾のうち、構造的な矛盾であるシーケンシンググラフと分岐条件の不足/重複の検証機能を実装した。また、学習シーケンシングより教材モジュールの提示の順序関係を抽出し、条件付き有向グラフとして視覚化して提示する機能も実装した。図9に妥当性検証器の実行例を示す。視覚化したグラフが画面左側に提示される。画面右側の表に、各辺に付加されている条件が表示されている。画面上部のボタンを押すことで検証アルゴリズムをグラフに適用し、結果が再描画される。

評価

学習シーケンシングの視覚化を行うことで、教材作成者の勘違いなどによる提示順の設計ミスの発見をサポートできるようになった。また、シーケンシンググラフの矛盾と分岐条件の不足/重複の矛盾の検出を実現した。これにより学習シーケンシング設計の際に提示順序の妥当性を保障できるようになった。ECA ルールを Prolog で記述したが、可読性の観点から難点が

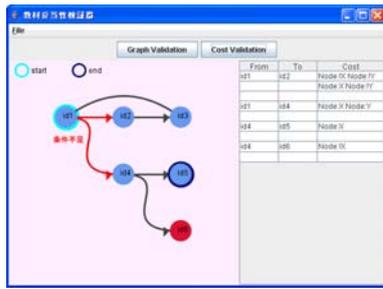


図9 実行例

あるため、上位レベルの記述を今後考えていく必要がある。

5. おわりに

本稿では、学習シーケンシングの妥当性検証において、教材の提示順序を抽出し条件付き有向グラフとして扱うことが有効であることを示した。条件付き有向グラフのモデル上から矛盾を定義し、一般的なグラフ探索のアルゴリズムを検証に利用できることを示した。また、学習シーケンシンググラフの構造を視覚化して提示することを実現した。

今後の課題としては、学習シーケンシングの矛盾についての再考察、未提案の検証方法について考察、矛盾の検証手法の改良、未実装の検証機能の早期実現が挙げられる。

学習上意味のある教材の提示順という観点で見れば、学習シーケンシングの意味的な矛盾は、本稿で定義した以外にも様々な場合があると考えられる。今後は意味のある提示順についてより考察を深め、検出の対象となる矛盾の定義の拡張を目指す。また、本稿で提案した分岐条件の検証手法は、条件がテストに合格、不合格などの2値で表現できるような場合でしか有効に働かない。今後はテストの合計得点や解答時間、学習履歴といった種々の学習者情報を組み合わせた複雑な条件の検証方法の提案も行っていく。文献[7]では、テストの評価結果は単純な正誤パターンではなく、問題の分野ごとの意味的なまとまりで正規化して扱うことを提案している。検証においても正規化した成績に対応することは重要な課題であると言える。また、教材作成者がECAルールを編集し、学習シーケンシングの設計を行うことは現実的には難しい。今後は教材作成者がGUIベースで直観的に学習シーケンシングの設計を行えるようなグラフオーサリングツールを開発する予定である。

文 献

- [1] SCORM2004, <http://www.adlnet.org/scorm/>.
- [2] IMS SS, “IMS Simple Sequencing Information and Behavior Model,” Version 1.0 Final Specification, <http://www.imsglobal.org/simplesequencing/>.
- [3] Paul De Bra, Ad Aerts, Bart Berden, Barend de Lange, Brendan Rousseau, Tomi Santic, David Smits, Natalia Stash, “AHA! The Adaptive Hypermedia Architecture,” Proceedings of the ACM Hypertext Conference, Nottingham, UK, August 2003.
- [4] 島内剛一, 有澤誠, 野下浩平, 浜田穂積, 伏見正則, “アルゴリズム辞典,” 共立出版, 1994年.
- [5] 延原哲也, 庄司成臣, 劉渤江, 横田一正, “ECAルールを活用したeラーニングシステムにおけるシーケンシング制御の改善,” DBSJ Letters, vol.4(no.2):pp.81-84, 2005.
- [6] 延原哲也, 劉渤江, 横田一正, “ECAルールによる適応型eラー

- ニング・シーケンシング制御モデル,” DBWS2006.
 [7] 西輝之, 延原哲也, 劉渤江, 国島丈生, 横田一正, “適応型eラーニングに必要な診断的テスト機構,” DBWS2006.