

SCORM-LST: SCORM による e ラーニングシステムの記述

SCORM-LST: Description Method of E-Learning Systems Using SCORM

森本 康彦^{1),2)} 植野 真臣²⁾ 柴田 晋吾³⁾ 横山 節雄⁴⁾ 宮寺 庸造⁴⁾

Yasuhiko Morimoto^{1),2)}, Maomi Ueno²⁾, Shingo Shibata³⁾ Setsuo Yokoyama⁴⁾, and Youzou Miyadera⁴⁾

¹⁾ 富士常葉大学 ²⁾ 長岡技術科学大学 ³⁾ コンパック株式会社 ⁴⁾ 東京学芸大学

¹⁾ Fuji Tokoha University, ²⁾ Nagaoka University of Technology, ³⁾ COMPAC, Ltd., ⁴⁾ Tokyo Gakugei University

1. はじめに

学校の教室等で実施される授業の多くは、教師と学習者、もしくは学習者と学習者の相互作用（以下、学習者のインタラクション）で成り立っている。しかしながら、現状の e-Learning システム（以下、LMS: Learning Management System）は、WBT(Web based Training)に見られるように、事前に用意されたコンテンツを中心とした学習が対象となっており、教師が意図した授業を行うことは、LMS の仕様で制限されるため容易なことではなく、教師は LMS に準じた授業運営を強いられることになる。

本研究の目的は、e-Learning システム上で、教師が意図する、学習者のインタラクションを中心とした授業（以下、クラスルーム的な授業）を実現することである。この授業は、教師もしくはシステムが個々の学習者の学習状態を把握し、適応的に学習者の行動を何らかの方法でコントロールしながら展開されるものでなければならない。この目的を達成するためには、教師の授業の設計に基づいて動作する LMS が必要となり、さらに、LMS の動作の基となる授業を構成する学習者のインタラクションと教師のコントロールを基礎とした学習者の学習状態遷移を形式的に記述する手法が必要とされる。そこで、本稿は、学習者のインタラクションと教師のコントロールによる学習状態遷移を形式的に記述する手法を開発し、この形式的記述の枠組みを SCORM2004[1]（以下、SCORM）に追加することで SCORM の拡張を行った(SCORM-LST)。

SCORM-LST は、LMS 開発者が、教師が意図する授業設計を理解するための手段ともなり得るため、教師は本記述の枠組みに基づいた授業設計を行うことで、教師の設計に応じて動作する LMS が期待できる。つまり、学習者のインタラクションを把握し、e-Learning 上でクラスルーム的な授業の展開が可能になると考えられる。

2. 記述の枠組みの要件

クラスルーム的な授業を e-Learning 上で実現させるためには、以下を要件とする新しい形式的記述の枠組みに基づいて動作する LMS が必要であると考えられる。

- 計算機（ここでは、LMS）で解釈が可能で、矛盾なく正確な記述ができる。
- 学習者のインタラクションが記述できる。
- 教師のコントロールが記述できる。

そこで、本稿では、学習者のインタラクションと教師のコントロールの記述に、学習状態遷移図を導入する（要件(b)と(c)に対応）。また、学習状態遷移のイメージを理解するだけでなく、それを計算機が解釈可能にするために、学習状態遷移図を記述するためのメタ言語を開発する（要件(a)に対応）。さらに、本稿では、上記要件を満たす記述の枠組

みを、SCORM 内に追加する(SCORM-LST)。

3. 学習者の状態遷移の記述

3.1. 学習状態遷移図 (LSTD)

学習者のインタラクションと教師のコントロールによる学習状態遷移を記述する学習状態遷移図 (LSTD: Learning State Transition Diagram) を提案する。

LSTD は、学習者のインタラクションと教師のコントロールによる学習者の状態遷移を有限オートマトンの理論を拡張し、遷移図として表現したものである[2][3]。図1の左上の LSTD は、問題演習を行う授業の例である。図中の○は「学習状態」、→と付随のラベルは「学習者のアクション」による学習状態遷移、●→と付随のラベルは「教師のアクション」による学習状態遷移、●→のみの表記は「システムのアクション」による学習状態遷移、◇は「教師もしくはシステムのコントロール」を表している。

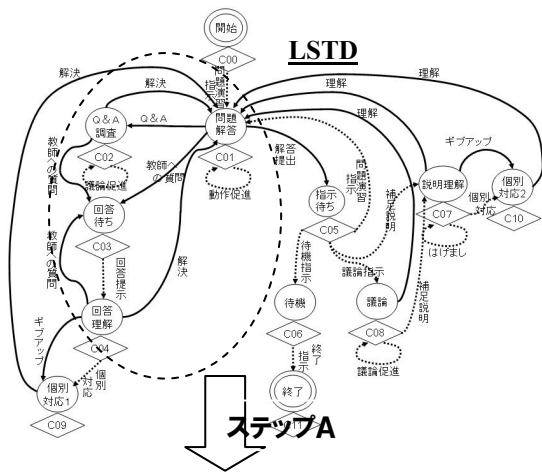
3.2. LSTD を記述するためのメタ言語

LSTD を計算機で扱うことができるように、LSTD を形式的に記述するためのメタ言語を提案する[2][3]。本メタ言語を用いることで、LSTD を矛盾なく正確に記述でき、さらに計算機が解釈可能になり、機械的に扱うことを容易にする。このメタ言語内では、状態遷移関数 T により、学習者の学習状態と学習者のアクションによって起きる状態遷移、さらに、教師のコントロールによるシステムの動作の記述法が定義されている。また、この記述方法に従うことで、各アクションやコントロールを容易に追加・変更が可能で、柔軟な LSTD の記述を可能にしている。図1左中に、メタ言語による記述の一例を示す。本研究では、本メタ言語で書かれたファイルを LST ファイルと呼ぶ。

4. SCORM-LST

本節では、SCORM に、学習者のインタラクションと教師のコントロールによる学習状態遷移の記述の枠組みを追加し、SCORM-LST を開発する[4][5]。広く実績のある SCORM に本記述の枠組みを追加することで、既存の学習コンテンツを用いながら、クラスルーム的な授業の実現の可能性が期待できる。しかしながら、SCORM は国際標準の規格であり、この規格に従い多くのコンテンツ、LMS が開発され実際に使われている。よってこれを考慮し、SCORM との互換性を保証する。そのため、SCORM のオリジナルの記述には手を加えず、manifest ファイルの<organization>要素内に新たに<LearningStateTransition>要素を追加する（図2）。本拡張に対応しない LMS は、このタグを読み飛ばすことで、オリジナルな動作を保証することができる。

LST ファイル中の状態遷移関数 T の記述は<studentInteraction>要素に、教師のコントロールの記述は



```

<state identifier="ST02">
<title> Q&A 調査 </title>
<studentInteraction>
<transitionRule>
<studentAction action="教師への質問"/>
<transitionState identifier-ref="ST03"/>
</transitionRule>
<transitionRule>
<studentAction action="解決"/>
<transitionState identifier-ref="ST01"/>
</transitionRule>
</studentInteraction>
<teacherInteraction identifier="C02">
<controlRule>
<callFunction function="停滞" argument="600,"/>
<teacherAction identifier="動作促進"/>
</controlRule>
</teacherInteraction>
</state>
<state identifier="ST03">
<title> 回答待ち </title>
<studentInteraction>
<teacherInteraction identifier="C03">
<controlRule>
<callFunction function="回答準備"/>
<teacherAction action="回答提示"/>
<transitionState identifier-ref="ST04"/>
</controlRule>
</teacherInteraction>
</state>
<state identifier="ST04">
<title> 回答理解 </title>
<studentInteraction>
<transitionRule>
<studentAction action="教師への質問"/>
<transitionState identifier-ref="ST03"/>
</transitionRule>
<transitionRule>
<studentAction action="解決"/>
<transitionState identifier-ref="ST01"/>
</transitionRule>
<transitionRule>
<studentAction action="ギブアップ"/>
<transitionState identifier-ref="ST21"/>
</transitionRule>
</studentInteraction>
<teacherInteraction identifier="C04">
<controlRule>
<callFunction function="停滞" argument="600,"/>

```

図1 記述の流れ

<teacherInteraction>要素に対応している。また、学習者のアクションは<studentAction>要素，教師のアクションは<teacherAction>要素，システムのアクションは<systemAction>要素で指定する。<callFunction>要素は，教師もしくはシステムのコントロールを実行するための関数（コントロール関数）を指定する。コントロール関数は，教師もしくはシステムのコントロールにそれぞれ対応し，事前にLMS内に準備されるべき組み込み関数である。

ルの記述と対応させ記述する（図1右）。これが，SCORM-LSTファイルとなる。

6. まとめ

本稿では，e-Learning上で，クラスルーム的な授業を実現させることを目的に，SCORMに学習者のインタラクションによる学習状態遷移を記述する枠組みを追加したSCORM-LSTを開発した。

現在，著者らは，SCORM-LSTに基づいて動作するLMSの開発をAcademicWare WBT V3.0[6]を拡張することで行っている。この開発により，SCORM規格に準拠したコースを用いた学習をユーザに対して保証しつつ，学習者のインタラクションと教師のコントロールを中心としたクラスルーム的な授業の実現が期待できる。

今後は，開発したLMSの実用を通し，SCORM-LSTとそのLMSの評価を行う。

7. 参考文献

- [1] ADLnet, *Sharable Content Object Reference Model: SCORM2004 2nd Ed.*, <http://www.adlnet.org/>, 2004.
- [2] 高橋正行, 森本康彦, 植野真臣, 横山節雄, 宮寺庸造, e-Learningにおける学習者の行動に基づく授業モデル記述法の提案, 信学技報 ET2005-4, pp.19-24, 2005.
- [3] Morimoto, Y., Ueno, M., Yokoyama, S. and Miyadera, Y., A Formal Method of Describing E-Language Systems, *Proc. Cognition and Exploratory Learning in Digital Age 2005*, pp.367-370, 2005.
- [4] Morimoto, Y., Ueno, M., Shibata S., Yokoyama, S. and Miyadera, Y., SCORM-LST: Describing Learning State Transitions Based on Learners' Interactions A Formal Method of Describing E-Language Systems, *Proc. The 2006 International Conference on SCORM2004*, 2006.
- [5] 森本康彦, 植野真臣, 柴田晋吾, 横山節雄, 宮寺庸造, 学習者行動に基づく学習状態遷移記述のためのSCORMの拡張, 信学技報, 2006. (2006年1月ET発表)
- [6] AcademicWare WBT V3.0, <http://www.compac.co.jp/SCORM2004/index.htm>, 2005.

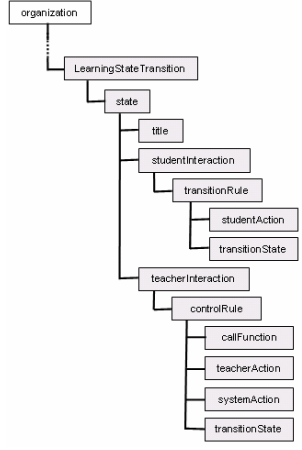


図2 <LearningStateTransition>要素のスキーマ

5. 記述の流れ

最初に，教師はLSTDを作成する（図1左上）。次に（ステップA），LSTDをメタ言語を用いてLSTファイルに表す。（図1左中）。さらに（ステップB），SCORMのmanifestファイルに<LearningStateTransition>要素を追加し，LSTファイ