

cmi5 対応学習管理システムの設計と開発

森本 容介^{*1}, 仲林 清^{*2}, 星野 忠明^{*3}, 前田 宏^{*4}

^{*1} 放送大学教養学部, ^{*2} 千葉工業大学情報科学部,

^{*3} エスエイティーティー株式会社, ^{*4} 株式会社ジンジャーアップ

Design and Implementation of the cmi5 Specification on a Learning Management System

Yosuke Morimoto^{*1}, Kiyoshi Nakabayashi^{*2}, Tadaaki Hoshino^{*3}, Hiroshi Maeda^{*4}

^{*1} Faculty of Liberal Arts, The Open University of Japan,

^{*2} Faculty of Information and Computer Science, Chiba Institute of Technology,

^{*3} SATT, Inc., ^{*4} GINGER APP Company

eラーニングコンテンツの新しい規格である cmi5 が注目されている。cmi5 のコース (コンテンツ) は、学習者に配信される教材やサービスの単位である Assignable Unit (AU) を階層に配置した構造を持つ。AU は、各種のデータを Experience API (xAPI) を用いて Learning Record Store (LRS) に記録する。REST アーキテクチャスタイルに基づく xAPI を採用したことにより、Web ページのフレーム内での動作を前提としない多様な教材が実現できる。これらの特徴を持つ cmi5 の実装に当たっては、学習管理システム (LMS) と LRS の連携や、AU 間遷移の方法を定める必要がある。本研究では、cmi5 対応 LMS の開発に当たり設計が必要な箇所を明らかにし、その設計を行った。LRS は、仮想的に LMS の背後に設置することにより、LRS の独立性を保ったまま cmi5 対応が行える。また、LMS にコンテンツの「目次」機能を設けることにより、学習者の指定による、または自動的な AU 間遷移が実現できる。これらの設計に基づき、Moodle の cmi5 対応プラグインを開発した。

キーワード: cmi5, Experience API (xAPI), 学習管理システム, Learning Record Store (LRS), eラーニング技術標準化

1. はじめに

eラーニングコンテンツの新しい規格である cmi5 が注目されている。以前から使われてきた規格である SCORM は、以下のような制約、問題がある。

- コンテンツが Web ページのフレーム内で動作し、かつ学習管理システム (以下, “LMS”) と常時通信できることを前提としている。
- 独習型のコンテンツしか実現できない。
- コンテンツの動作や、扱う学習履歴データの種類などが規格で定められており、拡張性が欠如している。様々な要求に対応すべく、規格が肥大化している。

- 学習履歴が LMS 内に閉じている。学習者が取り出すことや、他の種類の学習履歴と組み合わせて分析を行うためには工夫が必要である。

cmi5 は、これらを解決することを目的として策定された規格である。cmi5 では、学習者に配信され、トラッキング (学習履歴記録) の単位となる教材やサービスを、Assignable Unit (以下, “AU”) と呼ぶ。通信に、REST アーキテクチャスタイルに基づく Experience API (以下, “xAPI”) を採用したことにより、シミュレーションソフトウェアや携帯端末のソフトウェア (アプリ) など、Web ベースではない AU にも対応した。cmi5 では、AU の起動や終了時の動作を定めているが、AU 内における学習方法や、用いるデ

ータは規定しておらず、独習型以外の AU も容易に実現できると想定される。学習履歴は、xAPI を用いて、LMS とは独立した Learning Record Store (以下、“LRS”) に記録される。xAPI では学習履歴を取り出す方法も標準化されており、学習者による管理や、他の種類の学習履歴と組み合わせた分析などが可能である。

cmi5 は平易な規格であるが、実装に当たっては、LMS と LRS との連携方法や、AU 間遷移の方法を設計しなければならない。本研究では、LMS を cmi5 に対応させるための課題を明らかにし、システム設計、および開発を行うことを目的とする。

以下、2 章で、cmi5 の位置づけと、cmi5 対応 LMS の開発に当たって設計が必要な箇所を述べる。3 章で、cmi5 の規格を本稿に必要な範囲で述べる。4 章で設計を実施し、5 章で設計に基づき開発したシステムについて述べる。6 章で本研究の成果を考察し、7 章で本研究をまとめる。

2. cmi5 の位置づけと実装上の課題

2.1 xAPI と cmi5

e ラーニングシステム・コンテンツの標準規格として、2000 年代初頭から SCORM⁽¹⁾⁽²⁾が広く使われてきた。その後の情報技術の進展や、学習形態の多様化に対応する必要性が、2010 年前後に認識されはじめた。SCORM を主導した Advanced Distributed Learning (以下、“ADL”) は、Experience API (以下、“xAPI”) の策定を進め、2013 年にバージョン 1.0.0 をリリースした⁽³⁾。本稿執筆時点の最新バージョンは、2016 年にリリースされた 1.0.3 である。xAPI は、学習経験 (learning experience) の記録と交換を目的とした標準規格である。xAPI は、JSON で記述されたステートメント (statement) と呼ばれる構造で学習履歴を表現する。ステートメントが xAPI における学習履歴の記録単位であり、LRS に格納される。ステートメントのほか、学習活動 (教材) に関するデータを表現するステート (state)、学習者に関するデータを表現するエージェントプロフィール (agent profile) などが定められている。これらはいずれも JSON で記述され、LRS との間で送受信される。LRS との通信には、REST

アーキテクチャスタイルに基づく API が用いられる。そのため、学習活動の形態は Web ベースの教材に限らず、スマートフォンのアプリなど、各種のソフトウェアにも対応できる。

xAPI は、JSON で表現する各種データが従うべき規則を定めているが、その内容、つまり学習履歴の粒度や使用語彙は規定していない。それらの設計は利用者 (community of practice) に任されており、設計した内容はプロフィール (profile) と呼ばれる⁽⁴⁾。相互運用性の向上や実践の効率化を考えると、利用者独自の語彙やプロフィールを作成するのではなく、広く受け入れられたものを用いることが望ましい。そのようなプロフィールの作成には、実践の共有やそれに基づく改良が欠かせない。これまでに語彙やプロフィールを共有するレジストリが構築されている⁽⁵⁾⁽⁶⁾ほか、各種プロフィールの策定や、実践を通じた考察が行われている⁽⁷⁾⁽⁸⁾⁽⁹⁾。

2.2 cmi5 の特徴と実装上の課題

xAPI は、学習履歴をはじめとする各種データの送受信に特化しており、学習活動に関しては、ほとんど制約を設けていない。SCORM を対象とした xAPI のプロフィールも ADL によって策定されている⁽¹⁰⁾ように、xAPI は SCORM を置き換えるものではない。一方、LMS から配信されるコンテンツに関する規格として、cmi5 の策定が進められ、2016 年に正式版 (Quartz - 1st Edition) がリリースされた⁽¹¹⁾。cmi5 は SCORM に代わる規格として注目を集めている。

cmi5 は、xAPI のプロフィールの 1 つと位置づけられており、LMS から配信されるコンテンツの構造と、xAPI の使用規則を定めた規格である。前述の通り、学習者に配信され、トラッキングの単位となる教材やサービスを AU と呼ぶ。AU を階層構造に配置したものが cmi5 のコンテンツの単位であり、コースと呼ぶ。AU は、トラッキングデータをはじめとする各種のデータを、xAPI を用いて LRS に記録、または LRS から取得する。cmi5 が規定する範囲は SCORM と比較して少なく、簡易化されている。xAPI を採用したこととも相まって、cmi5 は多様な形態の AU や、学習環境に対応できる。

cmi5 コースの動作には LRS が必要であるが、現状

では多くの LMS は LRS の機能を持っていない。そのため既存の LMS を cmi5 に対応させる場合、LRS 機能を用意しなければならない。既存の LRS を用いる場合には、LMS と LRS の連携方法を定める必要がある。また、配信する AU を学習者が選択、または LMS が決定する方法も規定されていないため、多様な形態をとりうる AU の情報を集約した「目次」に相当する機能を持つインタフェースが必要である。

なお、cmi5 では、LRS を LMS の一部と位置づけ、両者をまとめて LMS と定義しているが、本稿では、両者を区別して、LMS に LRS は含めない。仕様書上の LMS は、“広義の LMS” と表記する。

3. cmi5 の規格

本章では、cmi5 の規格を本稿に必要な範囲で述べる。

3.1 コースの構造と学習状態

LMS は、コンテンツの構造と学習状態を管理する必要がある。cmi5 のコースは階層構造を持ち、根ノードがコース、根ノード以外の内部ノードがブロック、葉ノードが AU である。AU は、メタデータの moveOn の値に従って、学習者が基準を満たした (sufficiently completed) かどうかを判定する。moveOn の値は、「ステートメント“Passed”が発行されたとき」、「“Passed”と“Completed”のいずれかが発行されたとき」、など 5 通りが定義されている。ブロック配下のすべての AU、またはコース内のすべての AU が基準を満たしたとき、そのブロック、またはコースを目的語 (xAPI における“Actor, Verb, Object”の“Object”) として、ステートメント“Satisfied”を発行する。このことから、cmi5 コースの各ノードは、基準を満たしたか否かの 2 値の学習状態を持つと見なすことができる。

3.2 AU の起動から終了まで

LMS と LRS、および AU と LRS のインタフェースを検討するため、AU の起動から終了までの動作を述べる。

AU を起動する手順は次の通りである (図 1)。

- ① 学習者が AU を選択する。
- ② LMS が LRS に起動データを記録する。
- ③ LMS が LRS にステートメント“Launched”を

発行する。

- ④ LMS が学習端末を起動 URL に遷移させる。
- ⑤ AU が、LRS への接続に必要な認可トークンを取得する。
- ⑥ AU が LRS から起動データと学習者の設定 (cmi5LearnerPreferences) を取得する。
- ⑦ AU が LRS にステートメント“Initialized”を発行する。

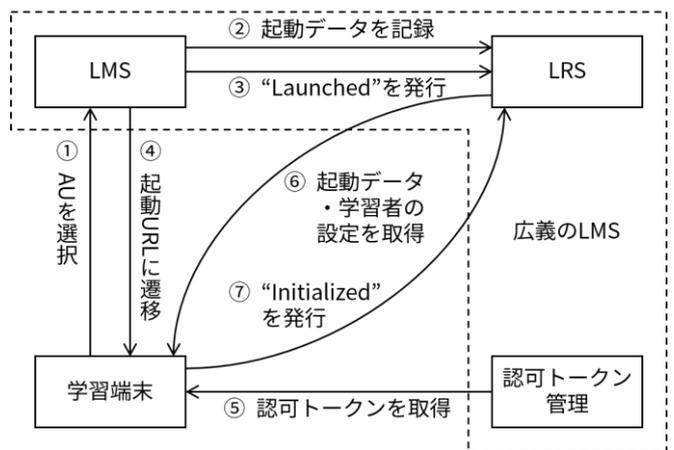


図 1 AU の起動手順

起動データ (launch data) とは、起動モード (launchMode)、合格点 (masteryScore)、AU 終了時の戻り先である returnUrl など、AU の動作を制御するデータである。起動 URL (launching URL、または launch URL) とは、AU の起動に用いられる URL である。起動 URL は、クエリ文字列として、学習者の識別子、LRS のエンドポイント、認可トークンの取得先などのデータを含む。認可トークンとは、AU が LRS に接続するために必要な、AU の起動から終了まで (AU セッション) の間だけ有効なトークンである。AU はすべての xAPI 通信に認可トークンを含め、LRS は渡されたトークンをもとに通信の可否を判断する。AU は起動 URL のクエリ文字列として渡されたデータをもとに認可トークンを取得 (⑤) し、LRS と通信 (⑥、⑦) する。なお、⑥と⑦の順は cmi5 の仕様書には明記されていないが、AU が完全に初期化されたことを示すために“Initialized”を発行すると定められているため、この順で差し支えないと考える。

AU セッションの間、AU は LRS に規格で定められたステートメントを発行するほか、規格に違反しない任意の xAPI 通信を行うことができる。

AU の終了時、AU は LRS にステートメント “Terminated” を発行する。起動データに returnURL が含まれていれば、AU は学習端末を returnURL にアクセスさせる。

3.3 学習者による AU の選択と AU 間の遷移

SCORM 2004 では、学習者がアクティビティ（階層構造を持ったコンテンツのノード）を選択できるインタフェースを、LMS が提供することが定められている。次のアクティビティに進む、コンテンツを終了する、といった動作を行うインタフェースも、LMS が提供する。また、学習者の要求に応じて、次に配信するアクティビティを決定する規則も規格で定められている。一方、cmi5 では、学習者に AU を選択させる方法や、「次」の AU を決定する規則などは定められていない。仕様書では、AU の順序は、コースの構造によって暗黙的に示されるとされており、先頭から順に学習することを想定していると考えられる。

4. cmi5 対応の設計

4.1 LRS の配置

本研究においては、既存の LMS に cmi5 を実装することを前提とする。cmi5 対応の LMS は、xAPI に完全準拠した LRS 機能を持たなければならないが、一般的に使われている LMS は LRS 機能を備えていない。そのため、cmi5 に対応させるためには、LRS 機能を追加実装するか、既存の LRS 製品を使用する必要がある。xAPI に対応した LRS 製品はすでに複数種類が開発、提供されているため、本研究では、既存の LRS を使った設計を行う。つまり、既存の LMS と LRS を別々に設置し、それらを連携させて cmi5 対応を行う。

4.2 LMS と LRS の連携方式

3.2 節においては、“Launched” は LMS が LRS に発行すると解釈しているが、仕様書では、広義の LMS が発行すると定めている。同様のステートメントに、“Abandoned”、“Waived”、“Satisfied”がある。これらのステートメントを、LMS と LRS のどちらが発行するかを定める必要がある。

また、3.2 節で述べた通り、AU は LRS と xAPI 通信を行う。ここで、AU と LRS が直接通信する構成と

する場合、以下の課題を解決しなければならない。

認可トークンの生成と共有

上記のステートメントを LMS が発行するなら、LMS が LRS に接続するための認可トークンが必要である。図 1 の“認可トークン管理”は、実際は LMS と LRS のどちらかが担当する必要がある。LMS が認可トークンを管理する場合、LRS との共有方法を定める必要がある。LRS が認可トークンを管理する場合、LMS との共有方法に加えて、LMS が持つ学習者や AU の情報を LRS が取得する方法の設計も必要である。なお、後述の通り、LMS と LRS の接続に用いる認可トークンは固定でも問題ないと考えられるが、AU 用の認可トークンを固定することはできない。LMS と LRS のどちらが管理するとしても、両者の情報共有が必要である。

LRS から LMS への通知

“Abandoned”は、起動中の AU が “Terminated” を発行する前に、同じコースの異なる AU からの xAPI 通信が行われた際に発行すべきステートメントである。“Abandoned”を発行すべき条件を満たしたかを判断できるのは、LRS である。“Abandoned”を LMS が発行するなら、規定の条件を満たしたことを LRS が LMS に通知しなければならず、その方法を定める必要がある。“Abandoned”を LRS が発行するとしても、異常終了が起こったことを LMS に通知する必要があるかを検討する必要がある。

以上のように、AU と LRS が直接通信する構成とする場合、LMS と LRS が密結合になる欠点がある。また、LRS は xAPI に準拠するだけでなく、cmi5 への対応も行わなければならない。

これらの問題を解決するため、図 2 のように、AU が LMS を通して LRS に接続する構成とする。AU には、LRS のエンドポイントとして、LMS 側に設置する通信中継サービスの URL を通知する。AU と LMS との間では、AU セッション内でのみ有効な認可トークンを用いる。この認可トークンは、LMS が管理する。LRS は LMS を信頼することを前提に、両者の xAPI 通信に用いる認可トークンを固定し、あらかじめ共有

しておく。LRS は単なるデータベースとして用い、
 “Abandoned” を発行する条件の判定など、cmi5 特有の機能は LMS に実装する。“Launched”、“Abandoned”、“Waived”、“Satisfied” の 4 種類のステートメントは、いずれも LMS が発行する。以上の方式を採用することにより、LRS は xAPI に対応しているだけでよく、cmi5 特有の機能を追加実装する必要がない利点を得られる。また、使用する LRS の種別変更も容易である。

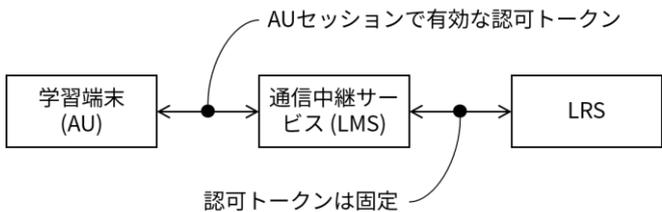


図 2 LMS と LRS の連携方式

ただし、認可トークンは HTTP リクエストメッセージの Authorization ヘッダを用いて送受信される。本設計を採用すると、LMS 自体に HTTP 認証⁽¹²⁾ (Basic 認証や Digest 認証) を設ける場合、Web サーバソフトウェア (httpd) の設定などに工夫が必要となる。

4.3 コースの目次とページ遷移

3.3 節で述べた通り、cmi5 では、学習者に AU を選択させる方法や、次に配信する AU を決定する規則は定めていない。SCORM をはじめとする一般的なコンテンツ同様、LMS が cmi5 コースの目次を表示し、学習者に選択させることが妥当と考えられる。一方、1 つの AU から構成されるコースや、順番に学習することを想定しているコースなど、目次が不要なコースも考えられる。そこで、目次の有無をコースごとに選択できることとし、次のようなインタフェース、およびページ遷移を設計した。

目次を設ける場合 (図 3)

起動する cmi5 コースを選択する画面をコースリストと呼ぶ。学習者がコースリスト上のコースを選択すると、コースの階層構造に従った目次を表示する (図 3 ①)。学習者が目次に表示された AU を選択すると、AU を起動する (図 3 ②)。このとき、returnURL として目次ページを指定する。学習者が AU を終了する

と、AU は returnURL の指定に従い、学習端末を目次ページに遷移させる (図 3 ③)。目次ページには、コースを終了するためのインタフェースを設ける。学習者がコースの終了を選択すると、コースリストに遷移させる (図 3 ④)。なお、目次ページでは、各ノードに、3.1 節で述べた学習状態を表示する。

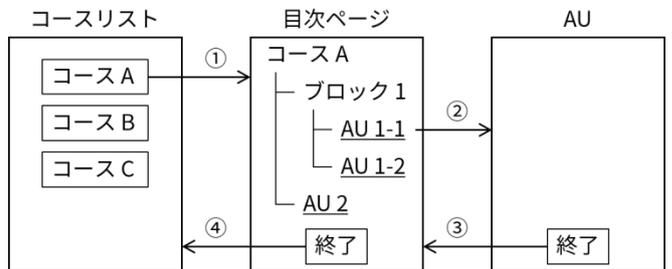


図 3 目次を設ける場合

目次を設けない場合 (図 4)

図 3 における目次ページに相当するページが、学習端末を AU、またはコースリストに自動遷移 (リダイレクト) させることにより、目次のないコースが実現できる。学習者がコースリスト上のコースを選択すると (図 4 ①)、リダイレクト用ページは最初の AU を起動する (図 4 ②)。学習者が AU を終了すると、リダイレクト用ページに戻る (図 4 ③)。終了した AU が最後の AU でない場合は、次の AU を起動する (図 4 ④→②)。最後の AU である場合は、コースを終了し、コースリストに遷移させる (図 4 ④')。

なお、AU の順は、マニフェストファイル (cmi5 のコース定義ファイル) に登場する順とする。目次を設けない場合、学習者が AU を終了したとき、AU だけを終了したいのか、コースも終了したいのかを区別できない。本設計では AU だけの終了と解釈するため、コース内で最後ではない AU の終了と同時に、コースも終了させることはできない。

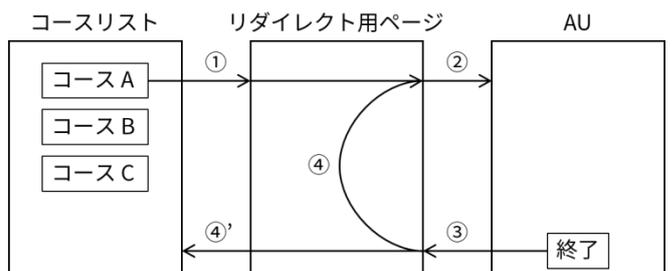


図 4 目次を設けない場合

5. Moodle と ELECOA を用いた実装

前章での設計に基づき、Moodle の cmi5 対応を行った。本章では、開発したシステムについて述べる。

5.1 ELECOA の概要と cmi5 への適用

著者らは、拡張性を有する学習支援システムのアーキテクチャである ELECOA の提案を行っている⁽¹³⁾⁽¹⁴⁾。ELECOA は、教材オブジェクトと呼ぶプログラム部品により、eラーニングシステムの機能拡張を行える。ELECOA の枠組みの上で新規機能を持った教材を動作させたいときは、その機能を教材オブジェクトとして実装すればよい。ELECOA は、SCORM のような階層型のコンテンツを対象としており、各ノードに教材オブジェクトを対応づける。cmi5 のコースも階層構造を持つことから、ELECOA との親和性が高い。cmi5 に対応した教材オブジェクト群を開発すれば、ELECOA の枠組みの上で cmi5 コースを動作させられる。

5.2 システム構成

本実装では、LMS として Moodle を用いる。これまでに、ELECOA に対応した Moodle のプラグインである mod_elecoa を開発した⁽¹⁵⁾。mod_elecoa は教材オブジェクトが動作するプラットフォームの役割を持ち、ELECOA の規格に従ったコンテンツを動作させられる。本実装では、Moodle の cmi5 対応に、mod_elecoa を用いる。つまり、cmi5 対応の教材オブジェクト群を開発することにより、mod_elecoa を通して、Moodle 上で cmi5 コースを動作させる。

mod_elecoa は、Moodle のプラグインの一種である活動モジュールである。掲示板 (mod_forum) や SCORM パッケージ (mod_scorm) 同様、cmi5 コースを Moodle コース上に活動として設置できる。図 3、および図 4 のコースリストは、Moodle のコースページとなる。図 3 の目次ページ、および図 4 のリダイレクト用ページは、mod_elecoa が提供する。図 2 に示した xAPI の通信中継サービスも、mod_elecoa が提供する。図 5 に、本システムの構成を示す。

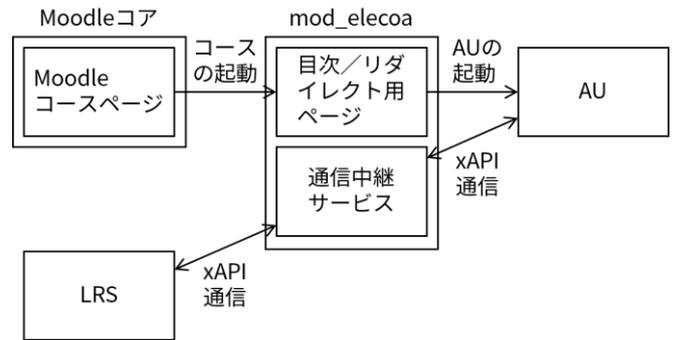


図 5 開発したシステムの構成

5.3 動作画面と環境

図 6 に、開発したプラグインが表示する目次画面を示す。これは、テスト用に開発した、7 つの AU からなるコースの目次であり、うち 2 つの AU に基準を満たしたことを示すチェックマークが表示されている。



図 6 cmi5 コースの目次

コースの起動から終了までの一連の流れが想定通り動作すること、LMS と LRS が連携動作し LRS に学習履歴が正しく記録されることなどをはじめ、本システムが正常に動作することを確認した。また、スマートフォンのアプリとして実装した AU を持つコースを制作し、目次画面からその AU を選択するとアプリが起動し、学習履歴も正しく記録されることを確認した。

なお、本実装では、LMS として Moodle 3.5 (本稿執筆時点においてリリース前の開発版)、LRS として Learning Locker を用いて動作検証を行った。LRS は、xAPI に準拠していれば、任意の製品に置き換えられることが期待される。

6. 考察

本研究で実施した **cmi5** の規格調査, および設計, 開発において得られた考察を述べる.

6.1 **cmi5** で実現できる教材について

cmi5 は, **SCORM** と比較して規格が簡易化されており, コースや **AU** に課せられる制約も少ない. **SCORM** の **SCO** (**cmi5** の **AU** に相当) は **Web** ページのフレーム内に配信されることを前提に, 制御用のスクリプトを **LMS** が提供する. 一方, **cmi5** の **AU** は, **LMS** の制御を受けることなく動作する. そのため, **AU** は **Web** ブラウザ上で動作する必要はない. たとえば, **AU** にソフトウェアを起動する役割を持たせたり, 起動 **URL** から直接スマートフォンのアプリを起動したりして, それらのソフトウェアを用いて学習させることができると考えられる. 本研究においても, 起動 **URL** にカスタム **URL** スキームを用いることにより, アプリ版 **AU** が起動し, 動作することを確認した. ただし, 以下に示す理由により, 今回開発したアプリ版 **AU** では, 終了後に目次ページに自動的に戻る動作を実装しておらず, 目次なしの設定では正しく動作しない. **cmi5** では, 同一コース内で, 複数の **AU** を同時に起動することは想定していない (同時起動を検知した場合, 異常終了を表すステートメント “**Abandoned**” を発行する). アプリの **AU** を使う場合や, **Web** ブラウザの新しいウィンドウで **AU** を起動する場合は, その **AU** を終了するまで, 目次ページで別の **AU** を選択できないようにしておくことが望ましい. また, **AU** の終了時に目次ページに戻る際, 元の目次ページを表示していたウィンドウを使う必要があり, かつ **AU** の状態を更新するための処理 (目次ページのリロードなど) も必要と考えられる. アプリの **AU** から目次ページに正しく戻ることが技術的に可能であるかは, 未検討である. このような新しいタイプの教材の動作検証を進める必要がある.

6.2 相互運用性の検証について

cmi5 は **SCORM** に代わる規格として注目を集めているが, 現時点では, 対応 **LMS**, オーサリングツールが十分提供されておらず, コンテンツもほとんど流通していない. そのため, **LMS** やコンテンツの動作に共

通理解が得られておらず, 相互運用性の検証が十分行われているとは言いがたい. 前章で述べた開発の際には, 動作検証に使用できる **cmi5** コースを出力するオーサリングツールが見つからず, テスト用コースの作成から行う必要があった. 規格の普及のためには, 各種ツールが提供され, 相互運用性が実証される必要があると考える. 本研究により, 最も広く使われている **LMS** である Moodle⁽¹⁶⁾ で **cmi5** コースを動作させることができるようになった. 本プラグインはオープンソースソフトウェアとして公開予定であり, 相互運用性の検証や, 規格の普及に役立てられることが期待される.

6.3 **LRS** の位置づけについて

cmi5 の規格では, **LMS** 自体が **LRS** の機能を有することが想定されている. しかし, 通常の **LMS** は独自のデータ永続化機能を有しており, **cmi5** に対応するために **xAPI** に準拠した **LRS** に切り替えることは現実的ではない. そもそも **xAPI** は, 多様なシステムの学習履歴を統合することを意図している. **LMS** 内に閉じた **LRS** より, 本研究における設計のように独立して設置した **LRS** を使う方が, 本来の目的に合致していると考えられる.

ただし, **LMS** 独自のデータと, **LRS** に記録されるデータが混在することになる. たとえば, Moodle の活動モジュールは, 起動, 終了などのイベントや, 学習成績などの学習履歴をデータベースに記録する. また, 「活動完了」という概念を持ち, 活動が完了の基準を満たしたかどうかをトラッキングできる. 一方, **cmi5** コースはこれらのデータを **LRS** に記録する. **LMS** と **LRS** に記録されるデータ間の関係を整理することは今後の課題である.

7. まとめ

本研究では, 既存の **LMS** と **LRS** を用いて **cmi5** コースを動作させるための課題を明らかにし, 設計を行った. **LMS** と **LRS** は別々に設置することができる. その際, 実装方法によっては, **LMS** が持つ情報を **LRS** で使用し, **LRS** で発生するイベントを **LMS** に通知する必要があるなど, 両者が密結合になる. **LMS** を通して **LRS** を使用することで, 複雑な実装を避け, **LRS**

の独立性を確保することができる。また、AU の情報を集約し、学習者に提示する「目次」の機能が必要であることから、本研究では、目次機能と AU 間の遷移の動作を定めた。

次に、本研究で実施した設計に基づいて、Moodle の cmi5 対応を行った。具体的には、Moodle の活動モジュールである mod_elecoa に、cmi5 対応機能を追加実装した。本プラグインは、相互運用性の検証や、規格の普及に役立てられることが期待される。

謝辞

本研究は、JSPS 科研費 17H00774, 17K18665 の助成を受けた。また、特定非営利活動法人日本イーラーニングコンソシアム技術委員会 xAPI 分科会技術検証 WG と連携して実施した。

参考文献

- (1) Advanced Distributed Learning: “SCORM 1.2 Specification”, http://www.adlnet.gov/wp-content/uploads/2011/07/SCORM_1_2_pdf.zip (2018 年 1 月 8 日確認)
- (2) Advanced Distributed Learning: “SCORM 2004 4th Edition Specification”, https://adlnet.gov/public/uploads/SCORM_2004_4ED_v1_1_Doc_Suite.zip (2018 年 1 月 8 日確認)
- (3) Advanced Distributed Learning: “Experience API 1.0.0”, <https://github.com/adlnet/xAPI-Spec/blob/1.0.0/xAPI.md> (2018 年 1 月 22 日確認)
- (4) Advanced Distributed Learning: “xAPI Profiles”, <https://github.com/adlnet/xapi-profiles> (2018 年 1 月 22 日確認)
- (5) The Registry, <https://registry.tincanapi.com/> (2018 年 1 月 22 日確認)
- (6) xAPI Vocabulary & Profile Index, <http://xapi.vocab.pub/> (2018 年 1 月 22 日確認)
- (7) 森本容介, 古川雅子, 山地一禎: “Moodle を対象とした xAPI のプロファイルの策定”, 情報処理学会研究報告, Vol.2017-CLE-21, No.20, pp.1-8 (2017)
- (8) Bakharia, A., Kitto, K., Pardo, A. et al.: “Recipe for Success - Lessons Learnt from Using xAPI within the Connected Learning Analytics Toolkit”, Proc. Sixth International Conference on Learning Analytics & Knowledge, pp.378-382 (2016)
- (9) Berg, A., Scheffel, M., Drachsler, H. et al.: “The Dutch xAPI Experience”, Proc. Sixth International Conference on Learning Analytics & Knowledge, pp.544-545 (2016)
- (10) Advanced Distributed Learning: “Experience API SCORM Profile”, <https://github.com/adlnet/xAPI-SCORM-Profile> (2018 年 1 月 23 日確認)
- (11) Advanced Distributed Learning: “cmi5 Specification Profile for xAPI”, https://github.com/AICC/CMi-5_Spec_Current/blob/quarterly/cmi5_spec.md (2018 年 1 月 8 日確認)
- (12) Fielding, R. T. and Reschke, J. F.: “Request for Comments 7235: Hypertext Transfer Protocol (HTTP/1.1): Authentication”, <https://tools.ietf.org/html/rfc7235> (2018 年 1 月 8 日確認)
- (13) 仲林清, 森本容介: “拡張性を有する適応型自己学習支援システムのためのオブジェクト指向アーキテクチャの設計と実装”, 教育システム情報学会誌, Vol.29, No.2, pp.97-109 (2012)
- (14) 仲林清, 森本容介: “拡張性を有する学習支援システムにおける再利用性向上のための教材オブジェクトデザインパターンの設計と実装”, 教育システム情報学会誌, Vol.35, No.3 (2018) (採録決定済み)
- (15) 森本容介, 仲林清, 芝崎順司: “ELECOA における教材オブジェクト・プラットフォーム間インタフェースの設計と実装”, 電子情報通信学会論文誌, Vol.J98-D, No.6, pp.1033-1046 (2015)
- (16) 高等教育機関における ICT の利活用に関する調査研究結果報告書 第 3 版, <https://axies.jp/ja/ict/2015report.pdf> (2018 年 1 月 24 日確認)