

# LTIによるLMSとのコース連携が可能な 多読能力推定テストの開発

田尻 大気<sup>1,a)</sup> 國宗 永佳<sup>2,b)</sup> 新村 正明<sup>1,c)</sup>

**概要:** 多読学習の効果向上のために適した本を学習者が選択する支援として、多読を推進する公益財団ERFは多読レベル推定テストシステムERFPTを提供している。ERFPTは独立したアプリケーションとして公開されており、世界24ヶ国の教育機関において授業の一環として利用されている。近年、高等教育機関においてはLMSが広く使われており、学習ツールをLMSプラグイン化することでLMS利用者はLMSのアカウントのみで様々な学習ツールを利用できる。しかしLMS毎にプラグインを作成しなければならず、LMSに組み込めるアプリケーションに制限があり難しい。一方で独立システムとして実装した場合、アカウントが重複することになり利用者の負担が大きい。学習ツールとLMSの連携を目的とした標準規格であるLTIでは、独立システムをLMSと連携できるためこれらの問題を解決できる。本研究ではLTIを使用してERFPTのテスト実施機能と結果閲覧機能をLMSから利用可能にする。

## 1. 背景

多読とは文章を分析せず大意を把握する読書法であり、絵本などのやさしい本から始め、年間100万語単位の読書量を目指す。多読ではグレイディッド・リーダー (GR) と呼ばれる、レベル毎に文章の難易度を調整した学習用書籍を使用する。しかしGRは出版社ごとに異なるスケールによってレベル分けされている。Extensive Reading Foundation (ERF) は教育機関において多読の推進を目的に活動する公益財団であり、その活動の1つとして出版社ごとに異なるGRを相互に比較可能にすることを目的とした、ERF グレイディッド・リーダースケール (ERF スケール) の策定をしている。

ERFでは学習者が自身のERFレベルを把握し多読に役立てるため、多読レベル推定テストシステムERF Placement Test (ERFPT) を公開している。ERFPTは当初、信州大学で開発され信州大学のみで利用されていたが、ERFに移管され現在では世界24ヶ国の教育機関において授業の一環として利用されている。

## 2. ERFPT

ERFPTは独立したWebアプリケーションとして構築

されており、テスト実施機能と結果閲覧機能およびアカウント管理機能を持つ。

### 2.1 テスト実施機能

ERFPTのテストは適応型テストになっており、テストで使用する問題セットはGRの書籍から抜粋した1, 2ページ程度の短い物語とその理解度を問う複数の正誤問題からなる。1回のテストは複数のラウンドからなり、1つのラウンドは、まず物語を読み、次にその理解度を問う設問に答える形式になっている。受講者の回答状況により異なる問題セットが選択され3から5ラウンド繰り返し、その結果により推定したレベルを受講者に提示する。またERFPTの問題セットは問題作成者により随時追加されている。

### 2.2 結果閲覧機能

ERFPTは授業で利用することを想定して開発されている。したがって受講者への多読能力推定機能に加え、教授者が自身の担当する受講者の学習状況を把握するため、担当する受講者のテスト結果を閲覧する機能を提供している。テスト結果では、推定されたレベル、読書にかかった時間、回答の正誤などの詳細が確認できる。

### 2.3 アカウント管理機能

ERFPTは単独のシステムとして構築されているため、専用のアカウントが必要である。通常ログインとゲストログインがあり、ゲストログインではその都度学籍番号など

<sup>1</sup> 信州大学大学院理工学系研究科

<sup>2</sup> 信州大学工学部

a) tajiri@seclab.shinshu-u.ac.jp

b) kunimune@cs.shinshu-u.ac.jp

c) niimura@cs.shinshu-u.ac.jp

の必要項目を入力することで、一度きりのアカウントとして利用できる。アカウントには、管理者、教授者、受講者のロールがあり、受講者ロールでは、アカウント作成および編集時に担当教授者を設定できる。認証方式はパスワード方式および OAuth 認証 (Twitter, Facebook, Google) が利用できる。

### 3. LMS と学習ツールの連携

近年の高等教育機関においては Learning Management System (LMS) が普及しており、コース管理の役割を担っている。また LMS がカバーできない特定分野の学習支援についてはそれに特化した学習ツールが導入されている。

#### 3.1 LMS プラグインと独立システム

学習ツールは LMS プラグインとして実装することで LMS に統合可能である。しかし LMS 毎にプラグインを作成しなければならないため開発コストが課題となる。また UI 上の制限を持たせたいものや外部システムとの連携が必要なものなど、特殊な条件を持つ学習ツールを LMS プラグインとして移植することは難しい。さらにコンテンツが頻繁に更新されるものなども、LMS 側でのプラグインの再インストールが必要になるため難しい。

一方で独立システムとして実装した場合はこれらの制約を回避できるが、アカウント管理およびその機能が LMS と重複することになる。したがって受講者に余計なアカウント作成、管理の手間、および教授者にそれをサポートする負担を強いることになる。さらに学習ツールの構築においてもコストが必要以上に掛かることになる。学習ツールでは本来コアとなる学習支援機能が重要であり、開発においてはその部分に注力すべきである。しかし単独のシステムとして開発する場合、学習支援機能以外のユーザ管理やコース管理などの一般的な機能を実装しなければならない。

#### 3.2 LTI

Learning Tools Interoperability (LTI) は外部ツールを LMS と統合可能にすることを目的とした標準規格である。LTI により、LMS プラグインを利用する際と同様に独立したシステムをログインし直すことなくシームレスに利用できる。LTI は Moodle や Blackboard など主要な LMS でサポートされており [1], LMS 毎にプラグインを作成しなくてよい。また HTTP を使用して連携を行うため、外部ツールは独立した Web システムでよい。そのため LMS プラグイン化の問題と独立システムにおけるアカウント管理の重複の問題を解決できる。また LMS との連携を念頭に策定されているため、アカウント情報だけでなくコース情報なども取得できる。

#### 3.2.1 LTI の詳細

ここでは LTI 1.0 [2] で策定されている Basic LTI について説明する。

LTI では学習ツールを提供するシステムを Tool Provider (TP) と呼び、学習ツールを利用するシステムを Tool Consumer (TC) と呼ぶ。TC は一般に LMS が相当する。またコースに準ずる、共通のユーザとロールを持つコレクションをコンテキストと呼び、コンテキストはコース、プロジェクト、グループなど複数のタイプがある。

教授者は TP の管理者からツールを利用するための資格情報を取得し、その資格情報を使用して TC のコース上に Launch Link と呼ばれるリンクを設置できる。コース上のユーザは Launch Link をクリックし学習ツールを利用できる。その際 Basic LTI Launch Data (Launch Data) と呼ばれる POST データが TC からユーザの Web ブラウザを経由し TP へ送信される。TP は Launch の際に、Launch Data に付与されているパラメータからユーザ、ロール、コースなどの情報を取得できる。

#### 3.2.2 LTI を使用した学習ツール

藤井らはプログラミング教育支援のシステムを LTI により LMS と連携させ、JUnit を用いた提出プログラムの自動テスト機能を実現している [3]。

学習ツールが授業で利用される場合、教授者が受講者の学習状況を把握するための機能も必要である。しかし先行研究では受講者への学習支援の機能に主に着目しており、それらの機能は提供されていない。

## 4. 提案

ERFPT は独立したシステムとして実装されているため、LMS と別のアカウントが必要となり受講者と教授者に過剰なアカウント管理の負担を課している。したがって ERFPT のテスト実施機能と結果閲覧機能を LTI を使用して LMS から利用可能にすることでこれを解決する。また開発コストの低減のため、テスト実施機能と結果閲覧機能に絞って、アカウント管理機能を廃す。

### 4.1 設計概要

ERFPT の 3 つの機能のうち、テスト実施機能、結果閲覧機能の 2 つを LTI 経由で利用できるよう修正を行い、これらの機能で必要とされていたアカウント管理機能の一部を廃す。

#### 4.1.1 コース管理

結果閲覧機能は教授者が自身の担当する受講者のテスト結果を閲覧する機能であり、今回提案するシステムでも同様の機能を実現するため、コース毎に閲覧できるよう実装する必要がある。ただし、ERFPT のコース管理については、コースの作成、コース上のメンバの管理、コース上のアイテムの利用制限といったコース管理機能は LMS が有

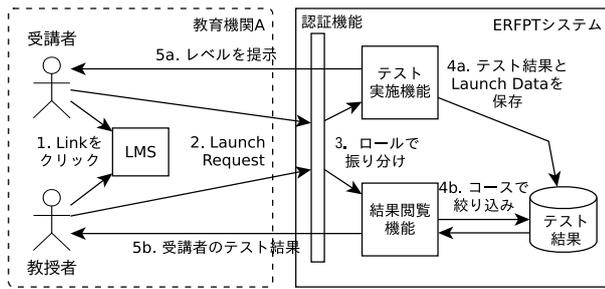


図1 システム構成

していると仮定し、これらの機能はLMSに移譲しERFPTでは実装しない。そのためユーザやコースといったテーブルは作成せず、Launch Dataを単なる値のセットとして単一のテーブルに保存する。

#### 4.1.2 結果閲覧機能

LTI 1.1 [4]ではLTI Basic Outcomes Service (Outcomes Service)により評点をLMSにフィードバックできるが、評点は0.0から1.0の数値でしか表現できない。ERFPTでは授業での支援に役立てるため、受講者の多読レベルや読書速度などテスト記録の確認する必要がある、評点ではこれらを表現できない。したがってOutcomes Serviceは使用せず、教授者もLaunch LinkからERFPTにアクセスし、ERFPT上で結果閲覧をすることとした。LTIではLaunchした際にLMSから送信されるLaunch Dataによってコース情報が得られる。このコース情報を使用して教授者がコース毎に受講生のテスト結果を閲覧できるようにする。

### 4.2 詳細

システムの概観を図1に示す。ユーザはLMSからLTIを経由して利用し、受講者にはテスト実施機能を、教授者にはテスト結果閲覧機能を提供するようロールによる振り分けを行う。具体的なシナリオは以下になる。

- (1) ユーザはコース上のLaunch Linkをクリックする。
- (2) ユーザのブラウザからERFPTにLaunch Dataが送信される。

ユーザが受講者であった場合、以下の経路に分岐する(図1上部)。

- (3a) ERFPTはLaunch Dataに付与されたロールから受講者だと判定しテスト画面を表示する。
- (4a) 受講者はテストを受験し、受験終了時、ERFPTは保持していたLaunch Dataとテスト結果を保存する。
- (5a) ERFPTは受講者に推定した多読レベルを提示する。

ユーザが教授者であった場合、以下の経路に分岐する(図1下部)。

- (3b) ERFPTはLaunch Dataに付与されたロールから教授者だと判定し結果閲覧画面を表示する。

(4b) ERFPTは結果閲覧画面を表示する際にLaunch Dataのコース情報によって絞り込みを行う。

(5b) ERFPTは教授者が担当するコースの受講者のテスト結果のみを表示する。

#### 4.2.1 ロールによる表示ページの振り分け

LTI経由でアクセスしたユーザのLMSにおけるロールは、Launch Dataのrolesパラメータによって識別できる。今回は教授者と受講者のみがLTI経由で利用すると仮定し、rolesパラメータの値がInstructorだった場合を教授者ロールに、それ以外を受講者ロールにマッピングした。そして教授者ロールの場合は結果閲覧ページに、受講者ロールの場合はテスト受験ページにリダイレクトするようにした。

#### 4.2.2 コースの絞り込み

教授者が結果閲覧画面にアクセスした際、結果閲覧機能では教授者が所属するコースに属する受講者のテスト結果のみを絞り込んで表示している。これには教授者と受講者のテスト結果がどのコースに属するものが識別できている必要がある。そのためにまず受講者の受験完了時にLaunch Dataを受験結果とともにDBに保存する。Launch Dataに含まれるパラメータのうち、tool\_consumer\_instance\_guidはTCのユニークIDで、context\_idはそのTCにおける起動したリンクを含むコンテキストのユニークIDである。したがってtool\_consumer\_instance\_guidとcontext\_idの2つのパラメータの組を識別子とすることでコースを一意に識別できる。

### 5. 評価

構築したシステムを信州大学の英語教員が受け持つ3つのクラスA, B, Cに導入し、複数の教育機関で利用されることを想定した実証実験を行った。ここでクラスは教育機関における教授者と受講者のまとまりを指し、コースはLMSにおけるまとまりを指す。クラスA, Bは信州大学生を対象としたクラスで、信州大学のMoodleに用意されたコースをそれぞれ利用している。クラスCは外部生を対象としたクラスで、通常の授業ではLMSは利用していない。

実験で使用したシステム構成を図2に示す。まず、実験用サーバを構築し、そのサーバ上にERFPTを構築した。複数の教育機関で利用できることを確認するため、実験用サーバにMoodleを構築し、クラスCのためのコースを作成した。これを別機関のコースとみなす。信州大学Moodle上のコースA, Bおよび外部クラス用Moodle上のコースCそれぞれにLaunch Linkを設置し、各クラスの受講者に対し、ERFPTの受験を課した。

教授者がそれぞれのMoodleのそれぞれのコースから結果閲覧画面をLaunchし、教授者がLTI経由でコース毎に受講者のテスト結果が閲覧できることを確認した。また受講者のテスト結果が保存されていることから受講者がLTI

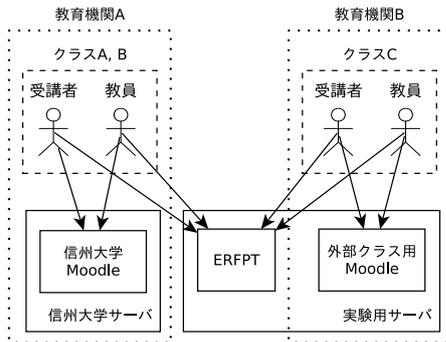


図 2 実験で使用したシステム構成

経由でテストを受験できているといえる。

一方で受講者の情報は受講者がテストの受験を完了しなければ ERFPPT に保存されないため、受験していない受講者のテスト結果はそもそもリストに現れない。これにより教授者が受験を完了していない受講者の発見が難しい問題が発生した。LMS 側ではコースの全ての受講者の評点のリストが見られることから、Outcome Service を使用して、受験者がテストを完了したかどうかを 1 か 0 で LMS にフィードバックすることで、これを解決可能であると考えられる。

## 6. 結言

本研究では LTI を利用し、コース管理機能の開発コストの低減および複数機関の利用を想定したコース単位の教授者のための受講者のテスト結果閲覧機能の実現を目的とし、システムの実装を行った。設計においては ERFPPT に求められる中心的な機能であるテスト実施機能と結果閲覧機能の既存の要求を満たしつつ、アカウント管理、コース管理を LMS に移譲することにより実装コストの削減を行った。そして構築したシステムを信州大学および信州大学外の 3 つのコースに導入し、そこで実証実験を行いコースごとにテスト結果が管理できることを確認した。

今回実験で使用した LMS は Moodle のみであった。他の LMS では細かい実装の差異により不具合が発生する可能性があるため、追加の検証を行う必要がある。

また ERFPPT のアカウント管理機能を廃し LTI 経由からのみ利用可能にしたため、LTI に対応した LMS を持たない教育機関では利用できない。しかし ERF では既に Moodle をホスティングし公開していることから、こちらに ERFPPT を起動する Launch Link を設置したコースを作成し利用させることで解決可能である。

今回結果閲覧画面では各受験結果の詳細情報を提示するに留め、コースに設置した Launch Link も 1 つのみだった。ただし、Launch Data には Launch Link の識別子となる `launch.link.id` パラメータが含まれているためコース上の各 Launch Link を識別可能である。したがってクラ

スで複数回受験させる場合、テスト毎に Launch Link を用意しそこから受講者に受験させることで何回目のテストを受験したかが識別できる。これによりテスト間の成長の可視化するといったことなども可能であり、今後はテスト分析などへの活用も検討する。

## 参考文献

- [1] IMS Global Learning Consortium: Product List, IMS Global Learning Consortium (online), available from (<https://www.imsglobal.org/conformance-list>) (accessed 2017-01-11).
- [2] IMS Global Learning Consortium: IMS Global Learning Tools Interoperability Basic LTI Implementation Guide v1.0, IMS Global Learning Consortium (online), available from (<http://www.imsglobal.org/specs/ltiv1p0/implementation-guide>) (accessed 2016-12-31).
- [3] 藤井聡一郎：LTI を利用した IT 基礎教育支援ツール max+ の開発，法政大学情報メディア教育研究センター研究報告，Vol. 29, pp. 83-86 (2015).
- [4] IMS Global Learning Consortium: IMS Global Learning Tools Interoperability Implementation Guide Final v1.1, IMS Global Learning Consortium (online), available from (<https://www.imsglobal.org/specs/ltiv1p1/implementation-guide>) (accessed 2017-02-18).