

Experience API に対応した学習成果物収集のための Web ブラウザ拡張機能の開発

Development of the Experience API Adopted Web Browser Extension for Gathering Learning Outcomes

米山基, 松葉龍一, 合田美子, 中野裕司
Motoi Yoneyama, Ryuichi Matsuba, Yoshiko Goda, Hiroshi Nakano
熊本大学大学院社会文化科学研究科教授システム学専攻
Graduate School of Instructional Systems, Kumamoto University
Email: myoneyama@st.gsis.kumamoto-u.ac.jp

あらまし：MOOC の隆盛に見られるよう、昨今人々の e ラーニングにおける学習スタイルが変容し、個人の学習の記録や成果物が複数の組織の LMS 等 Web 上の学習ツールに散在するということが珍しくなくなっている。このような状況に対応するため、本稿では Experience API に対応したブラウザ拡張機能を開発し、学習ツールの違いに関わりなく、ブラウザ上での様々な学習の成果物を効率的に一所に蓄積する方法を提案する。

キーワード：Experience API, ブラウザ拡張機能, 学習成果物, データ連携

1. はじめに

今日の MOOC の隆盛などに見られるように、学習者が自身の所属組織にとらわれず、LMS 等 Web 上のさまざまな学習ツールを利用する場面が多く見られるようになってきている。学習者にとってこのような状況には学習の機会が広がるというメリットがある一方、学習の履歴や、学習成果物のデータがそれぞれの学習ツール内に散在し、扱いにくくなってしまいう問題もある。これを解決するためには、複数の学習ツール内に蓄積されている情報を収集し、学習者がコントロールできるひとつの場所に保存するということが必要になる。そこで本研究では、学習に関する情報の収集、蓄積のための規格である Experience API を活用し、より多くの種類の情報を、学習ツールの区別無く効率的に収集するシステムを提案、開発する。

2. 先行事例

複数の学習ツール内のデータを連携して利用する試みについては、すでに以下のような方法をとる事例が存在している。

- ・ 特定の 2 種類以上の学習ツール同士それぞれを拡張して互いのデータを連携する方法をとるもの (e.g. Moodle と Mahara の連携)
- ・ ミドルウェアを開発し、それを通じて複数ツール間でデータをやり取りするという方法をとるもの (e.g. Glue!, Campus Project)

しかしこれらのうち前者はあくまで特定のツール間の連携のみしかできないという点、後者は各ツールのもつデータの構造の差異を吸収しきれず結果としてごくわずかな種類のデータしかやり取りできないという点で問題があった。

一方かねてより SCORM の後継として開発の進ん

でいた、学習情報の収集、蓄積に関する標準規格、Experience API の Version 1.0 が 2013 年にリリースされた。この規格を用いたシステムは、以下の 3 点でこれまでの事例の欠点を克服するシステムを構築しようものと考えられる。

- ・ 各学習ツールから収集した情報をツール間で直接連携させようとするのではなく、LRS (Learning Record Store) と呼ばれる共通の学習情報記録専用のシステムに保存し、それを相互利用させるというアーキテクチャを前提とする点
- ・ 各ツール内の学習成果に関する情報を「誰が何をどうした」というシンプルかつ柔軟性のある「ステートメント」という形式で取り扱うことにより、より多くの種類の学習情報に対応できるという点
- ・ 「ステートメント」に添付ファイルをつけることができるため、学習の履歴や評価のみならず、たとえば提出した課題のファイルや小テストの解答、ディスカッションのポスト等学習の結果生み出された成果物そのもののデータも蓄積できる点

3. Experience API の活用

現在 Experience API を活用している事例の多くは図 1 の通り、LMS や Web 上のコンテンツを Experience API に対応させ、そこから LRS にデータを送信するというアプローチをとっている。このような方法をとれば、当然 Experience API 対応済みツールであれば何であれ、そのデータを LRS に収集蓄積することができよう。しかし 2014 年 6 月現在では 80 近くの Experience API 対応製品が存在しているという報告もあるものの、まだ現状実稼働している

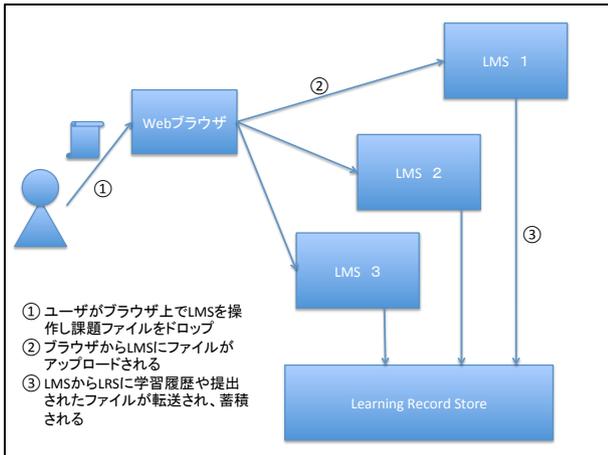


図 1 Experience API 対応製品の例

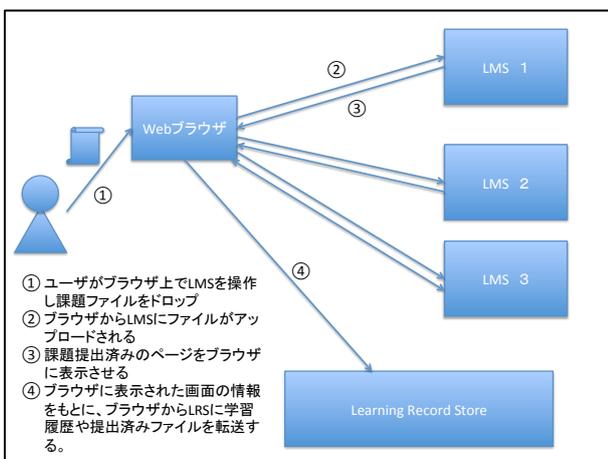


図 2 本研究の構想するシステム構成

学習ツールについては未対応のものが多く存在している。

そこで本研究では、学習ツール側の Experience API 対応状況に関わらずデータの収集を可能にするため、図 2 のようにユーザの使用するブラウザから直接 LRS に情報を送信する方法をとる。この方法をとる先行事例はすでに存在しているが (Klaas Poortinga 2014)、現状ブラウザで閲覧中の Web サイトの URL や単純なテキスト情報のみを扱うにすぎない。本研究ではユーザの学習成果物そのものの蓄積という点に着目し、これを実現するシステムを提案する。

4. ブラウザ拡張機能の動作

上述のシステムを実現するために、本研究では Web ブラウザ「Google Chrome」の拡張機能を開発し、LMS「Moodle」上の活動モジュールでの学習の結果、成果物をメタデータとともにステートメント化して LRS に送信する仕組みを構築した。開発したブラウザ拡張配下のような流れで動作する。

- ① あらかじめ拡張機能には LRS のアクセス先 URL、およびユーザがもつ LRS のアカウント情報を設定しておく。

- ② ブラウザが表示しているページの URL を常時監視。URL のパターンが Moodle の「課題」や「小テスト」の提出結果を表示するページと一致する場合は、データを LRS に転送可能なページであることをユーザに通知
- ③ ユーザがデータを LRS に送信することを選択した場合は、表示されているページの HTML をパースして、そこから得られた情報を使い、Experience API に準拠した JSON 形式の「ステートメント」を作成する。例えば「課題」モジュールの場合は、課題提出画面上に表示されている課題名、評点等を Experience API ステートメントの object や result プロパティの値として設定し、また実際に提出したファイルを画面上のリンクから取得して、その容量、ファイル名等を attachment プロパティの値として設定する。またファイルそのものも RFC1341 における multipart/mixed の定義に準拠してステートメントに添付する。
- ④ 作成したステートメントを、あらかじめ拡張機能にユーザによって設定された認証情報を使って BASIC 認証を通過し LRS 側に POST する。

なお、上記とは別に、今回開発対象外のモジュールや、Moodle 以外の学習ツール上のデータ収集に対応するために、ユーザが手動で入力したメタデータをもとに作成したステートメントに、ブラウザ上に表示されている HTML をそのままローカル保存したデータを添付して LRS に添付ファイルとして送信できる機能も実装した。

5. おわりに

今回の研究においては、利用する学習ツールの別なく、簡単にその学習情報を収集する仕組みを構築することを企図した。今後は、LRS に蓄積した情報をいかにして活用するかという点に注目し、今回開発したブラウザ拡張により蓄積した情報を Mahara 等 e ポートフォリオシステム上で活用する仕組みの開発を行う。

参考文献

- (1) 米山基, 中野裕司, 久保田真一郎, 合田美子: ”学習成果の観点からみた LMS・学習ツール間連携技術の比較検討”, 教育システム情報学会第 37 回全国大会(千葉工業大学)発表論文集, pp.116-117 (2012)
- (2) Advanced Distributed Learning: “Experience API”, <https://github.com/adlnet/xAPI-Spec/blob/master/xAPI.md> (参照 2014.06.22)
- (3) Rustici Software: “Who’s using the Tin Can API?”, <http://tincanapi.com/adopters/> (参照 2014.06.22)
- (4) Klaas Poortinga: “I did this!”, <https://chrome.google.com/webstore/detail/i-did-this/bnkehgbnfngelmfnbjicdcicafoegka?hl=ja> (参照 2014.06.22)