

Basic LTI による Moodle1.9 と Moodle2.3 との統合

王躍^{†1} 小柏香穂理^{†1} 刈谷丈治^{†2} 小河原加久治^{†1}

Basic LTI は、IMS Global Learning Consortium で策定されている LMS プラットフォーム間で学習ツールの相互運用性を高めるための規格である。本稿では、Basic LTI を用いた Moodle1.9 と Moodle2.3 との統合によって、Moodle1.9 のコースから Moodle2.3 の新しい機能を利用する方法を提案し、Moodle1.9 から Moodle2.3 への円滑な移行に役を立つことを期待したい。

Integrating Moodle1.9 with Moodle2.3 by Basic LTI

YUE WANG^{†1} KAHORI OGASAWA^{†1}
JOJI KARIYA^{†2} KAKUJI OGAWARA^{†1}

Basic LTI is a IMS Standard that allows web based learning tools to be easily integrated in LMS as native ones. This paper makes a proposal for Moodle1.9 interoperating with Moodle2.3 by using the existing modules that supports Basic LTI standard. Our approach will be a good warm-up to move to Moodle2.3 from Moodle1.9.

1. はじめに

Moodle は世界で最も広く利用されているオープンソース e ラーニングシステムの 1 つである[1]。2012 年の現時点で Moodle2.3 がリリースされ[2]、Moodle1.9 から大幅な機能強化を行ったが、Moodle1.9 と 2.3 のコース間の完全な互換性が保証されていないため、既存の Moodle1.9 のコースを単純に移行することができない。したがって、当面 Moodle 1.9 をも使用し続けていると思われる。本稿では、Basic LTI を用いた Moodle1.9 と Moodle2.3 との統合によって、Moodle1.9 のコースから Moodle2.3 の新しい機能を利用する方法を提案する。

2. IMS LTI について

LTI (Learning Tool Interoperability) は、IMS Global Learning Consortium で策定されている LMS (Learning Management System) プラットフォーム間で学習ツールの相互運用性を高めるための規格である[3]。図 1 に示すように、IMS LTI では、「Tool Consumer」が「Tool Provider」の提供する学習ツールを利用するために必要な、ユーザ認証やデータ連携・セッション管理などに関する技術の規格をまとめたものである。

現在 IMS LTI の規格は、LTI 1.0 (Basic LTI)、LTI1.1、LTI2.0 (策定中) という 3 種類のものがある。LTI1.1 と Basic LTI との違いは、「Tool Provider」から「Tool Consumer」への評価のフィードバック機能実装の有無にある。

3. Moodle と Basic LTI

Maria Jose Casany, Marc Alier and Jordi Piguillem が開発した Moodle1.9 のモジュール「basicliti」は、Basic LTI の「Tool Consumer」機能を実装している[4]。一方、Juan Leyva が開発した Moodle2.3 のローカルプラグイン「Itiprovider」は、Moodle2.3 のコース全体、もしくは、コース活動を Basic LTI の「Tool Provider」としての機能を実装している[5]。「basicliti」と「Itiprovider」を両方合わせて利用すれば、Moodle1.9 のコースから Moodle2.3 のコースや活動をシームレス (Single-Sign-On) に使える。つまり、Moodle1.9 のユーザは Moodle2.3 において実現された新しい機能を手軽に利用できることになる。そのためのシステムの構成は図 2 に示す。(ちなみに、Moodle2.2 以降では、コア活動モジュール「外部ツール (External tool)」が LTI Consumer として追加されている。)

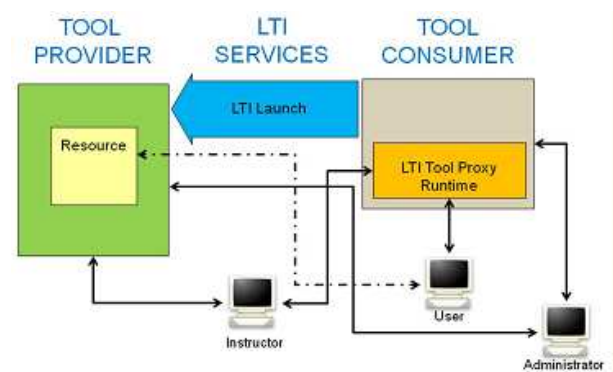


図 1. IMS LTI の概観 (文献 3 から引用)

Figure 1 Overview of IMS LTI [3].

^{†1} 山口大学・メディア基盤センター
Yamaguchi University, Media and Information Technology Center
^{†2} 山口大学・名誉教授
Yamaguchi University, emeritus professor

Moodle1.9 のモジュール「basicliti」は「<http://code.google.com>」

com/p/basiclti4moodle/ からダウンロードできる. そのインストールや利用方法などについては文献[6]を参照されたい.

また, Moodle2.3 のローカルプラグイン「ltiprovider」は, 「http://moodle.org/plugins/pluginversions.php?plugin=local_ltiprovider」からダウンロードできる. 一旦 Moodle2.3 にインストールされると, コースナビゲーション・ブロックに「LTI Provider」のリンクが表示される (図3参照). そのリンクをクリックすると, 同コースにおいて提供する LTI 「ツール」の追加や編集ができる (図4参照).

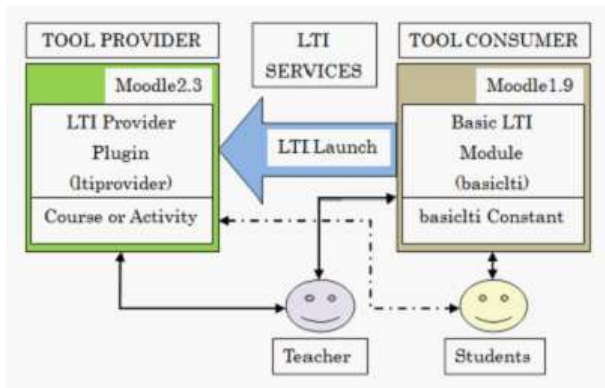


図2. 統合システムの構成

Figure 2 Structure of the integrating system.

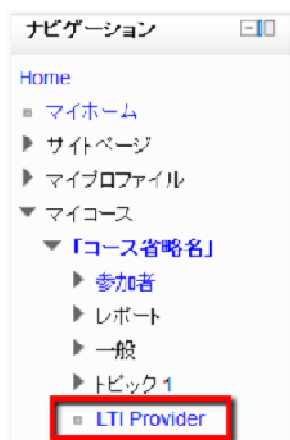


図3. 「LTI Provider」リンク

Figure 3 Link of "LTI Provider".



図4. 「LTI Provider」初期画面

Figure4 Opening screen of "LTI Provider".

4. 問題点

しかし, 公開されているプラグイン「basiclti」と「ltiprovider」をそのまま利用すると, 次の2つの不都合な

点がある.

(1) (Moodle1.9 サイト管理者が) 作成した LTI 「ツール」 [6] は Moodle1.9 のすべてのコースから (ランチャーURL を参照して) 利用できること.

(2) Moodle1.9 と Moodle2.3 のコース間の LTI 「ツール」の利用は, コースの教師権限だけでできないので, Moodle1.9 サイト管理者の権限も必要であること.

本稿では, 元のプラグイン「basiclti」と「ltiprovider」に対して, カスタマイズを行い, これらの問題を解決する方法を提示する.

5. プラグイン「ltiprovider」の改造

上述の問題点(1)について, すべてのコースからの利用を防ぐため, Basic LTI ランチャーデータ[3]のカスタムパラメータを利用して, LTI 「ツール」へのアクセス制限をかける. 具体的には, LTI 「ツール」を作成するとき, 「Tool Provider」と「Tool Consumer」との共有秘密鍵と, 同じようなパラメータ「プライベートキー (privatekey) 」 (データ型 = text) を保存するためのフィールドとして, プラグイン「ltiprovider」の DB テーブル「mdl_local_ltiprovider」に追加する (図5参照). そして「Tool Consumer」ランチャーデータのカスタムパラメータ「custom_privatekey」との照合によって, LTI 「ツール」へのアクセスの可否を決める. 照合が一致しない場合は LTI 「ツール」へのアクセス (ログイン) を拒否する (図6参照).



図5. Moodle2.3 における LTI 「ツール」の追加画面

Figure 5 Adding screen of "LTI Tool" in Moodle2.3.

改造されたソースファイルと追加した PHP コードは次の通りである.

(1) DB テーブル「mdl_local_ltiprovider」定義の変更: 「db/install.xml (19 行目)」を次の通りに編集する。(既に

元のプラグインをインストールされた場合、プラグイン「Moodle Adminer」を使ってDBテーブルの変更を行うことができる。

```
<FIELD NAME="secret" TYPE="text" NOTNULL="false"
SEQUENCE="false" PREVIOUS="arolelearn"
NEXT="privatekey"/>
<FIELD NAME="privatekey" TYPE="text"
NOTNULL="false" SEQUENCE="false" PREVIOUS="secret"
NEXT="encoding"/>
<FIELD NAME="encoding" TYPE="char" LENGTH="32"
NOTNULL="false" SEQUENCE="false"
PREVIOUS="privatekey" NEXT="institution"/>
```

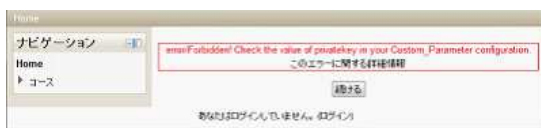


図6. 「プライベートキー」によるLTI「ツール」へのアクセス拒否

Figure 6 Deny access to "LTI Tool" by "private_key".

(2) 「プライベートキー」フォームの追加：
 「edit_form.php」(103行目)に次の行を追加する。

```
// =====
$form->addElement('text', 'privatekey', get_string('privatekey',
'local_ltiprovider'), 'maxlength="64" size="25"');
$form->setType('privatekey', PARAM_MULTILANG);
$form->setDefault('privatekey', md5(microtime()));
//ランダム初期値を与える
$form->addHelpButton('privatekey', 'privatekey',
'local_ltiprovider');
// =====
```

(3) 「プライベートキー」を照合する機能の追加：
 「tool.php」(43行目)に次の行を追加する。

```
// =====
$customsecret = $context->getCustomPrivatekey();
if ( !empty($tool->privatekey) && !($customsecret ==
$tool->privatekey) ) {
    print_error("Forbidden! Check the value of privatekey
in your Custom_Parameter configuration."); }
// =====
```

(4) ライブラリ関数の追加：
 「ims-blti/blti.php」(139行目)にカスタムパラメータから「プライベートキー」を取り出す関数を追加する。

```
// =====
```

```
function getCustomPrivatekey() {
    $customsecret = $this->info['custom_privatekey'];
    if (isset($customsecret) && strlen($customsecret) > 0 )
return $customsecret;
    return false;
}
// =====
```

(5) プラグインの日本語化など：
 プラグインの言語パック「lang」を編集する(詳細は省略)。

6. モジュール「basiclti」の改造

Moodle1.9のコースにおいて、教師権限でLTI「ツール」を活動として追加できるように、元のモジュール「basiclti」のソースファイルを次のように編集した。

(1) Moodle1.9コースの編集モードで「basiclti」のモジュール名(Basic LTI)を「活動の追加...」一覧に表示させるため、ソースファイル「lib.php」に定義されている関数「basiclti_get_types」を無効化する(図7参照)。

(2) モジュールのDBテーブル「mdl_basiclti」に、次の2つのフィールド(フィールド名, 種別, 照合順序, NULL, デフォルト値)を追加する。

- resourcekey, var(255), utf8_unicode_ci, YES, null
- password, var(255), utf8_unicode_ci, YES, null



図7. 「活動の追加...」一覧

Figure 7 List of "Add an activity...".

(3) ファイル「locallib.php」(58行目から)の関数「basiclti_view」のコード

```
// =====
$typeconfig = basiclti_get_type_config($instance->typeid);
$endpoint = $typeconfig['toolurl'];
$key = $typeconfig['resourcekey'];
$secret = $typeconfig['password'];
$org_id = $typeconfig['organizationid'];
// =====
を次のように編集する。
// =====
$typeconfig = (array)$instance;
```

```

$typeconfig['sendname'] = 1;
$typeconfig['sendemailaddr'] = 1;
$typeconfig['customparameters'] =
$instance->instructorcustomparameters;
$typeconfig['acceptgrades'] = 1;
$typeconfig['allowroster'] = 1;
$typeconfig['allowsetting'] = 1;
$endpoint = $instance->toolurl;
$key = $instance->resourcekey;
$secret = $instance->password;
$sorg_id = $instance->organizationid;
if ( ($instance->typeid) != 0 ) {
    $typeconfig =
    basiclti_get_type_config($instance->typeid);
    $endpoint = $typeconfig['toolurl'];
    $key = $typeconfig['resourcekey'];
    $secret = $typeconfig['password'];
    $sorg_id = $typeconfig['organizationid'];
}
// =====

```

(4) ファイル「mod_form.php」(56 行目) に次の行を追加する。

```

// =====
$this->typeid = 0;
$this->typeconfig = null;
// =====
また、入力フォームの項目「外部ツール URL (toourl)」、  

「コンシューマーキー (resourcekey)」、  

「共有秘密鍵 (password)」、  

「カスタムパラメータ (instructorcustomparameters)」の  

対応コードを次のように追加・編集する。
// =====
$regex = '/^(http|https):\/\/[a-z0-9-]+.*\/i';
$form->addElement('text', 'toolurl',
get_string('toolurl', 'basiclti'), array('size' => '64'));
$form->setType('toolurl', PARAM_TEXT);
$form->setHelpButton('toolurl', array('toolurl',
get_string('toolurl', 'basiclti'), 'basiclti'));
$form->addRule('toolurl', get_string('validurl', 'basiclti'), 'regex',
$regex, 'client');
$form->addRule('toolurl', null, 'required', null, 'client');
$form->addElement('text', 'resourcekey',
get_string('resourcekey', 'basiclti'));
$form->setType('resourcekey', PARAM_TEXT);
$form->addElement('passwordunmask', 'password',
get_string('password', 'basiclti'));
$form->setType('password', PARAM_TEXT);
$form->addElement('header', 'custom', get_string('custom',

```

```

'basiclti'));
$form->addElement('textarea', 'instructorcustomparameters', "",
array('rows' => 15, 'cols' => 60));
$form->setType('instructorcustomparameters',
PARAM_TEXT);
// =====

```

更に、入力フォームを簡単化するため、不要なオプションをすべて削除し、必要最小限の項目だけにした（図 8 参照）。これで Basic LTI に関する操作は、すべて教師権限で行うことができるようになり、上述の 2 つの問題点を解決した。なお、Moodle サーバにおいて、「php.ini」設定ファイルに「default_charset = "utf-8"」と設定する必要がある。さもなければ、ブラウザによって Basic LTI に関するエラー「Invalid signature」が発生することがある。



図 8. 活動「Basic LTI」の追加画面
 Figure 8 Opening screen of " Basic LTI " activity.

7. おわりに

本稿では、Web に公開されている Moodle プラグイン「basiclti」と「ltiprvider」を改造して、Basic LTI を用いた、Moodle コースの教師権限での Moodle1.9 と Moodle2.3 との統合を実現した。Moodle1.9 のコースの学生は、Moodle2.3 を意識することなく、Moodle2.3 の機能を利用することができるので、Moodle1.9 から Moodle2.3 への円滑な移行に役を立つことを期待したい。

参考文献

- 1) <http://moodle.org/about/> (accessed 2012-7-31)
- 2) http://docs.moodle.org/dev/Moodle_2.3_release_notes (accessed 2012-7-31)
- 3) IMS Global Learning Tools Interoperability Implementation Guide(Final Version 1.1.1), <http://www.imsglobal.org/lti/index.html> (accessed 2012-7-31)
- 4) BasicLTI4Moodle, <http://code.google.com/p/basiclti4moodle/> (accessed 2012-7-31)
- 5) Moodle-local_ltiprvider, https://github.com/jleyva/moodle-local_ltiprvider (accessed 2012-7-31)
- 6) Basic LTI for Moodle Instruction.pdf, <http://code.google.com/p/basiclti4moodle/downloads/list> (accessed 2012-7-31)