

Design a Moodle Synchronous Learning Activity

Lung-Hsing Kuo, Te-Hsiung Tseng, Hung-Jen Yang,
National Kaohsiung Normal University
No.116, Heping 1 st Rd., Lingya District, Kaohsiung City 802, TAIWAN(R.O.C.)
admi@nknuc.nknu.edu.tw, cedriccfc2002@gmail.com,
&
Hsieh-Hua Yang
Oriental Institute of Technology
No.58, Sec. 2, Sichuan Rd., Banqiao City, Taipei County 220, Taiwan
Republic of China
yansnow@gmail.com

Abstract: - Moodle is a famous open source Course Management System, but this system could only provide limit synchronous communication service through keyboard chatting. So the purpose of this study was to development a Moodle synchronous learning activity module to establish adobe video conference meeting. Based upon the Moodle developer guide and Adobe Acrobat Connect Pro server guide, this study designed a Moodle & Adobe Acrobat Connect Pro bridge unit and defined the correspondence between Moodle roles to Adobe Acrobat Connect Pro principals. Based on the proposed Module, Moodle really will use for full online classes as well as supplementing “face-to-face” learning and help students to improve their learning progress.

Key-Words: Moodle, Course Management System, Rich Internet applications, XML

1 Introduction

Technological developments in the design of Learning Management System are opening the way for exploitation in a wide range of applications. Many educational projects have promoted and introduced e-learning as an alternative to traditional education.[1] Information and Communication Technologies (ICT) with their continuous development provide new possibilities for the creation of innovative and effective environments of teaching and learning, thus redefining the educational processes.[2] With accelerated growth of computer and communication technologies, researchers have attempted to adopt computer network technology for research on education. Notable examples include the development of Course Management System. The website «www.cmsreview.com» provides directories with descriptions of over 350 Proprietary CMS, Open-Source CMS, and Hosted CMS services at application service providers.[3]

The Course Management System (CMS) alternatively known as a Learning Management System (LMS) or a Virtual Learning Environment (VLE). Moodle is a famous open source Course Management System. It is a Free web application that educators can use to establish effective online learning sites[4] Moodle system was designed with following characteristics.

1. It was designed to be used for full online classes as well as supplementing face-to-face learning

2. It was designed to be simple, lightweight, efficient, compatible, low-tech browser interface
3. It was designed to be easy to install on almost any platform that supports PHP. It requires only one database (and can share it).
4. It was designed to be full database abstraction supporting all major brands of database (except for initial table definition)
5. It was designed to show descriptions for every course on the server, including accessibility to guests with course lists.
6. It was designed to provide courses categories and courses searching functions. - one Moodle site can support thousands of courses
7. It was designed to emphasis on strong security throughout. Forms are all checked, data validated, cookies encrypted etc
8. It was designed to have most text entry areas (resources, forum postings etc) using an embedded WYSIWYG HTML editor [5]

This popular CMS system could only provide limit synchronous communication service through keyboard chatting. There is a need to implement more fruitful functions for synchronous communication, such as audio and video communication.

On the other hand, the web is evolving into machine-centric and user-centric perspectives. We are seeing tools and platforms to facilitate user collaboration and sharing collected under the term

"web 2.0" and influencing all types of communities.[6] Rich Internet applications (RIAs) are used into in a variety of web applications. The envisioned growing autonomy of such vehicles and the complexity of the environments in which they are expected to operate brings a new vision of the future role of internet applications. The Adobe flash application is a mature technology in all kinds of Rich Internet applications technologies. Acrobat Connect Pro is a online synchronous learning solutions base on flash for virtual classroom environments.

Most eLearning solutions require specialized software or plug-in downloads, but Acrobat Connect Pro delivers learners to the classroom with just a simple click—from anywhere, at anytime, on almost any device. Things user required is what they already have: a web browser and the ubiquitous Adobe Flash® Player software, already installed on 98% of Internet-connected computers.[7]

Acrobat Connect Pro is based upon an open documented and flexible architecture that provides money-spinning integration with existing infrastructure and future possible investments. Acrobat Connect Pro uses standards such as XML

and Java™ to exchange data, and it offers a full developed APIs and SDKs.[7]

Adobe Connect Pro meetings are live online conferences for compound users. A meeting room is an online application could be used to conduct a meeting. The meeting room consists of various display panels (pods) and components. Choosing from prebuilt meeting room layouts or customize a layout to suit your needs is possible. Users could apply meeting rooms to teach online classes and to collaborate with colleagues or students.[8]

Adobe Acrobat Connect Pro is a possible solution to be applied on Moodle for establishing synchronous learning activity, but there still is no Moodle module which could be used to establish video conference meeting based upon connect pro. So the purpose of this study was to development a Moodle synchronous learning activity module to establish adobe video conference meeting. Based upon the Moodle developer guide and adobe connect server guide, this study designed on-line learning activity to make the synchronous communication service possible for Moodle users. A prototyping method was applied to conduct this research.

Table 1 Program files of establishing an activity module

filename	Future
mod_form.php	a form to set up or update an instance of this module
version.php	defines some meta-info
icon.gif	a 16x16 icon for the module
index.php	a page to list all instances in a course
view.php	a page to view a particular instance
lib.php	any/all functions defined by the module should be in here
backuplib.php	for course backup or restore
restorelib.php	
settings.php	a definition of an admin settings page for this module
defaults.php	Lets you easily define default values for your configuration variables.
db/install.xml	Defines the structure of db tables for all database types. Is used during module installation
db/upgrade.php	Defines changes in the structure of db tables. Is used during module upgrade
db/access.php	defines module capabilities
lang/en_utf8/widget.php	Lastly, each module will have some language files that contain strings for that module.

2 Problem Formulation

The problem of this study was that there existed no audio and video synchronous communication service for the Moodle course management system. In order to study how to integrate Moodle with Adobe Acrobat Connect Pro, we have studied the development of their manual to

find out the functions they provide. Moodle R & D manuals can be obtained on the website (<http://docs.Moodle.org/en/Development>).

Adobe Acrobat Connect Pro's R & D manuals can also be obtained on the official website. Therefore, the prototype module could be created accordingly.

2.1 Moodle module directory structure and uses

The version of Moodle system in this study is 1.9x. Moodle system did provide well defined specifications. In Table 1, a list of required program files for an activity module was presented. The lib.php program file defines a number of critical functions.

These functions are listed in Table 2. All function names are with a prior naming part. For example, If the module name is called widget, to create a new instance of the function was called widget_add_instance() , to delete a exists

instance of the function was called widget_delete_instance()

There are also other important considerations in creating Moodle module:

When creating a new module, the new name of the module must not contain numbers or other special characters.

There need a data base table with the same name as the module. This table must have at least three fields : id , course , name

It is also required to make sure that the activity module provides appropriate support for groups and meta-courses[9]

Table 2 Functions of establishing an activity module

function name	Future
_install()	will be called during the installation of the module
_add_instance()	code to add a new instance of widget
_update_instance()	code to update an existing instance
_delete_instance()	code to delete an instance
_user_outline()	given an instance, return a summary of a user's contribution
_user_complete()	given an instance, print details of a user's contribution
_get_view_actions()	Used by the participation report to classify actions in the logs table
_delete_course()	code to clean up anything that would be leftover after all instances are deleted
_process_options()	code to pre-process the form data from module settings
_reset_course_form()	used to implement Reset course feature.
_delete_userdata()	

2.2 Roles operation of Moodle

In a Moodle system, the role attribute is processed by two primary functions:

1. define a list of permissions - role definition is global for all contexts, but can be changed by local context overrides
2. replace old course enrolments - role assignment in course context is similar to the old enrolment process

The new system will allow authorized users to define an arbitrary number of roles (eg a teacher)

A role consists of a list of permissions for different possible actions within Moodle (eg delete discussions, add activities etc).

Roles can be applied to users in a context. [10]There are some default roles in a Moodle system. Those roles have some basic rights:

1. Admins: Admin users will be assigned the default legacy admin role in the system (site) context

2. Teachers: Users who were teachers will be assigned the default legacy teacher role (or non-editing teacher role) in all courses they were teacher.
3. Students: Users who were students will be assigned the default student role in all courses they were student.
4. Guests: There will still be a single guest user with no default role at site level.[10]

2.3 Adobe Acrobat Connect Pro key concepts

Adobe Acrobat Connect Pro has three key concepts:

1. Principals : include users and groups
2. SCOs: Shareable Content Objects represent meetings, courses, and just about any content that can be created on Connect Enterprise.
3. Permissions: which define how principals can act on objects[11]

Table 3 four kinds of permissions [11]

Perr	Description
view	The principal can attend a meeting as participant, or view a folder's content.
host	The principal is host of a meeting
min	The principal is presenter of a meeting and can present content, share a screen, send text messages, moderate questions, create text notes, broadcast audio and video, and push content from web links.
rem	The principal does not have participant, presenter or host permission to attend the meeting.

2.4 Adobe Acrobat Connect Pro Meeting Permissions

There are four kinds of permissions in a Adobe Acrobat Connect Pro Meeting system, these four permissions are listed in Table 3

2.4 Adobe Acrobat Connect Pro Principals and Meetings Processes

There are many processes used to manage Principals and meetings in a synchronous communication activity, such as a Adobe Acrobat Connect Pro server. R & D in the adobe connect development manual describes a variety ways to manage accounts. It can be viewed as four parts comprising "User manager", "Shareable Content Objects (SCO) manager", "User Authentication" and "Meeting manager". (See Fig. 1)

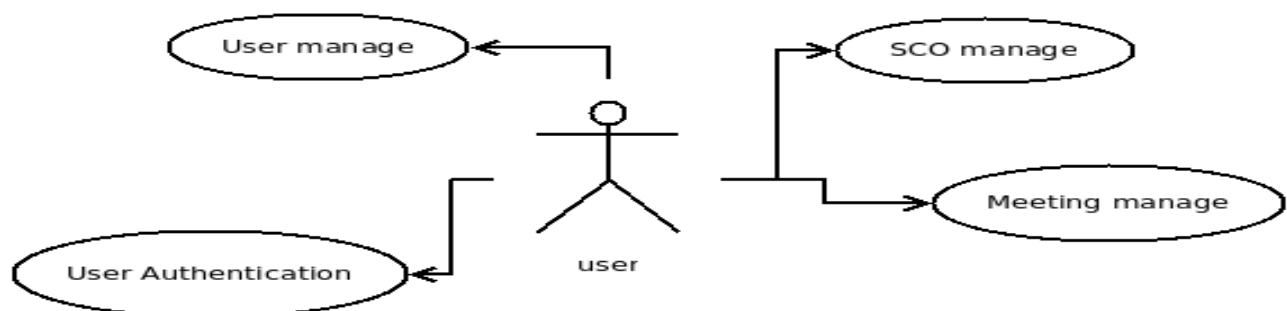


Fig.1 Adobe connect api user case

There are five events in "user manager" part:

1. Create a new user and send a welcome e-mail.
2. Create a new user without using an e-mail address as a login ID.
3. Update standard user information.

4. Update custom field values for a user.
5. Define a custom field and set it on a user.

These process steps of events showed in Table 4

Table 4 events of User manager

events	Process setps	Needed api
Create a new user and send a welcome e-mail	1. In your application, log in as an Administrator user. 2. Call principal-update with at least these parameters: https://example.com/api/xml?action=principal-update&first-name=jazz&last-name=doe&login=jazz99@doe.com&password=hello&type=user&send-email=true&has-children=0&email=jazz99@doe.com	Principal-update
Create a new user without using an e-mail address as a login ID	1. In Enterprise Manager, navigate to Administration > Users and Groups > Edit Login and Password Policies. Make sure that Use E-mail Address as the Login is set to No. 2. In your application, log in as an Administrator user. 3. Call principal-update to create the new user, passing both login and email parameters: 4. Parse the response for the principal-id of the new user:	Principal-update
Update standard user information	1. Log in as an Administrator user. 2. Call principal-list with a filter to get the user's principal-id . 3. Call principal-update to update the user 4. Parse the response for a status code of ok.	principal-list principal-update
Update custom field values for a user	1. Log in as an Administrator user. 2. Call custom-fields to get the field-id of the custom field: 3. Get the principal-id, sco-id, or account-id you want to update. 4. Call acl-field-update to update the value of the custom field:	custom-fields acl-field-update

There are two events in “User Authentication” part:

1. Log in using the session parameter

Table 5 events of User Authentication

events	Process setps	Needed api
<i>Log in using the session parameter</i>	1. Before you log the user in, call common-info to get the value of the BREEZESSESSION cookie.	common-info
	2. Extract the cookie value from the response.	
	3. Log the user in, specifying the cookie value.	login
	4. Parse the response for a status code of ok.	
	5. Use the session parameter with the same cookie value on subsequent calls for the user, until the user’s session ends.	
	6. When the user logs out or the user’s session ends, do not reuse the cookie value.	logout
Get the principal-id of a user or group	1. Call principal-list with a filter.	principal-list
	2. Parse the principal elements in the response for the principal-id.	

2. Get the principal-id of a user

These process steps of events showed in Table 5

There are six events in “SCO manager par”t:

1. Find a SCO when you do not know the sco-id
2. Get information about a SCO
3. Construct the URL to a SCO
4. Download a zip file from the server

5. Check the permission a principal has on a SCO
6. Check all principals’ permissions on a SCO

These process steps of events showed in Table 5

Table 6 events of SCO manager

events	Process setps	Needed api
Find a SCO when you do not know the sco-id	1. Call sco-shortcuts to get a list of root folders on Connect Enterprise:	sco-shortcuts
	2. Parse the response for a type of the root folder that would logically contain the SCO, for example, my-courses for a course the user has created.	
	3. Parse the resulting sco element for a sco-id:	
	4. Create a call to sco-expanded-contents to list the contents of the folder, adding an exact match filter, if possible:	
	5. Parse the response for the sco-id:	
Get information about a SCO	1. Call sco-info with the sco-id:	
	2. Parse the response for name, url-path, or any other value:	
Construct the URL to a SCO	1. Call sco-shortcuts:	sco-shortcuts
	2. Parse the response for the domain-name value in any sco element:	
	3. Call sco-info with the sco-id:	sco-info
	4. Parse the response for the url-path:	
	5. Concatenate the url-path with the domain-name:	
Download a zip file from the server	1. Call sco-shortcuts:	sco-shortcuts
	2. Extract any domain-name value from the response:	
	3. Call sco-info with the sco-id of the zip file:	sco-id
	4. Parse the response for the url-path element:	
	5. Construct the download URL, for example:	
Check the permission a principal has on a SCO	1. Call permissions-info with both an acl-id and principal-id:	permissions-info
	2. Parse the response for a permission-id:	
Check all principals’ permissions on a SCO	1. Call permissions-info with an acl-id, but no principal-id:	permissions-info
	2. Iterate through the principal elements and parse them for permission-id values:	

“Meeting manager” part can be viewed as five parts comprising “Create meetings”, “Find meetings”, “Create meeting room

URLs”, “Invite users to meetings” and “Remove users from meetings”. (see Fig. 2)

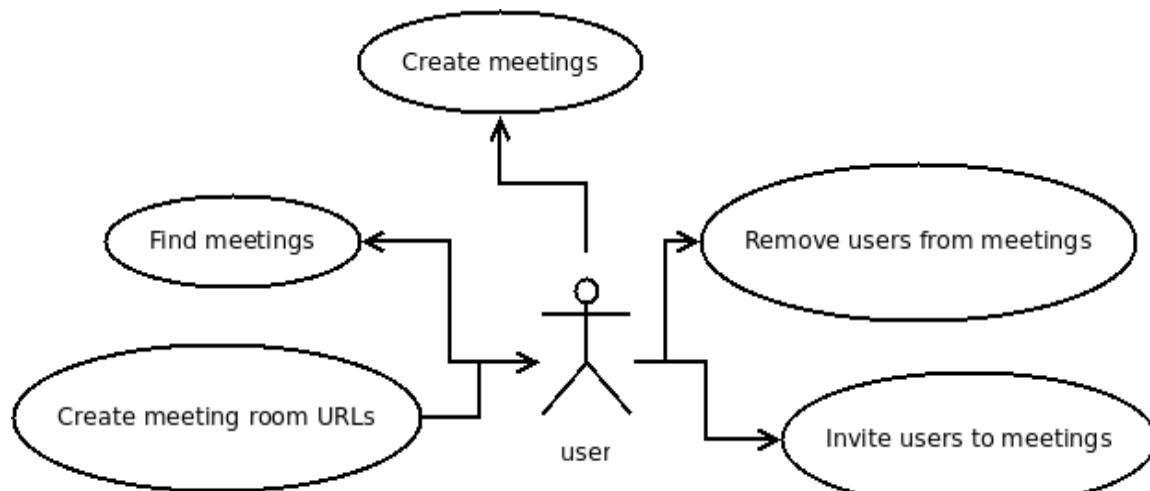


Fig 2 Meeting manager part

There are two events in “Create meetings” part:

1. Create a public meeting and add host, presenter, and participants
2. Create a private meeting and add host, presenter, and participants

These process steps of events showed in Table 7

Table 7 events of Create meetings

events	Process setps	Needed api
Create a public meeting and add host, presenter, and participants	1. Call principal-list to check that the user creating the Acrobat Connect meeting is a member of the live-admins group.	principal-list
	2. Call sco-shortcuts to obtain the sco-id of the user's my-meetings folder.	sco-shortcuts
	3. Parse the response for the sco element with type=my-meetings.	
	4. Call sco-update to create the meeting room.	sco-update
	5. Parse the response for the sco-id of the new meeting.	
	6. Call permissions-update to make the meeting public. Use the sco-id of the meeting as the acl-id.	permissions-upda te
	7. Call permissions-update to add a host, a presenter, and participants.	permissions-upda te
	8. Create the URL to the meeting room.	
Create a private meeting and add host, presenter, and participants	1. Log in as your application's Administrator user.	
	2. Follow the steps for creating a public meeting, but set the meeting ermission to private.	permissions-upda te
	3. Call permissions-update again to add a host, a presenter, and guests.	permissions-upda te
	4. Create the URL to the meeting room.	

There are four events in “Find meetings” part:

1. List all meetings on the server
 2. Display a user's meetings
 3. Display a user's meetings
 4. Add a user to a meeting and display meetings
- These process steps of events showed in Table 8

Table 8 events of Find meetings

events	Process setps	Needed api
Find the sco-id of a meeting	1. Call sco-shortcuts	sco-shortcuts
	2. Parse the response for the sco-id of a meetings folder that is likely to contain the meeting	
	3. Call sco-contents on the folder, adding a filter or two to reduce the response	sco-contents
	4. Parse the response for the sco-id of the meeting	
List all meetings on the server	1. Call report-bulk-objects with type=meeting.	report-bulk-objects
Display a user's meetings	1. Log the user in (see "Log in from an application").	
	2. Call report-my-meetings to list the user's meetings.	report-my-meetings
	3. Parse the response for values from the meeting elements.	
	4. Create the URL to the meeting room .	
Add a user to a meeting and display meetings	1. Log in as your application's Administrator user.	
	2. Get the user's principal-id.	
	3. Get the sco-id of the meeting.	
	4. Call permissions-update to add the user to the meeting.	permissions-update
	5. Log out as the Administrator user, and log in as the user you just added to the meeting.	
	6. Display the user's current meetings.	

There are two events in "Create meeting room URLs" part:

1. Create the URL to a meeting room for which the user is host

2. Create the URL to a meeting room for which the user is not host
These process steps of events showed in Table 9

Table 9 events of Create meeting room URLs

events	Process setps	Needed api
Create the URL to a meeting room for which the user is host	1. If you are logged in as a user, and you want to create a URL to a meeting in the user's my-folder, call report-my-meetings.	report-my-meetings
	2. Parse the response for the values of domain-name and url-path.	
	3. Concatenate the two values and add http:// or https:// at the beginning.	
Create the URL to a meeting room for which the user is not host	1. Get the sco-id of the meeting.	
	2. Call sco-info with the sco-id.	sco-info
	3. Parse the response for the url-path.	
	4. If you do not know the domain name, call common-info.	common-info
	5. Parse the response for the value of the host element.	

There are two events in "Invite users to meetings" and "Remove users from meetings" parts :

1. Send an e-mail to meeting participants.
2. Remove a user's permission to access a meeting.

These process steps of events showed in Table 10

Table 10 events of Invite users to meetings and Remove users from meetings

events	Process setps	Needed api
Send an e-mail to meeting participants	1. Call sco-info with the meeting sco-id.	sco-info
	2. Parse the response for the meeting name, date, or other values.	
	3. Construct the URL to the meeting room.	
	4. Call permissions-info to get the principal-id values of the presenters or participants, filtering on permission-id.	permissions-info
	5. Parse the response for the principal-id values you want.	
	6. Call principal-info with the principal-id.	principal-info
	7. Extract the name and email values from the response.	
	8. Call permissions-info again, filtering on a permission-id of host.	permissions-info
	9. Parse the response for the principal-id.	
	10. Call principal-info, using the principal-id.	principal-info
	11. Parse the principal element of the response for the name and login (or name and email).	
Remove a user's permission to access a meeting	1. (Optional) Call permissions-info to check the principal's permission to enter the Meeting;	permissions-info
	2. Get the meeting's sco-id	
	3. Get the user's principal-id (see "Find a principal-id").	
	4. Call permissions-update, using the meeting's sco-id as the acl-id and permission-id=remove:	Permissions-update

3 Problem Solution

3.1 Base Connect Unit

Adobe Connect Enterprise Web Services were designed to be used with multiple languages on multiple platforms which ever could send and receive XML over HTTP to develop custom applications.[8] We need to have there three tools:

1. XML parser code library
2. A cookie management code library
3. viewing HTTP request and response headers

There are three XML parser code libraries for php. Those are Document Object Model (DOM), XML Parser, XMLReader and SimpleXML. This study uses SimpleXML to parser XML because the extension of SimpleXML provides a very simple and easily usable toolset to convert XML to an object

which could be processed with normal property selectors and array iterations.

This study use fsockopen function to develop a HTTP Client class to send HTTP request and get its response for creating code library of cookies managing and HTTP request and response headers viewing. This class could be also used to send cookies in each http request.

For A cookie management code library and viewing HTTP request and response headers tools this study use fsockopen function to develop a HTTP client class to send http request and get HTTP response ,this class is also can send cookie in every http request.

On the other hand, this study designs a Moodle & Adobe Acrobat Connect Pro bridge unit illustrated in Fig.3:

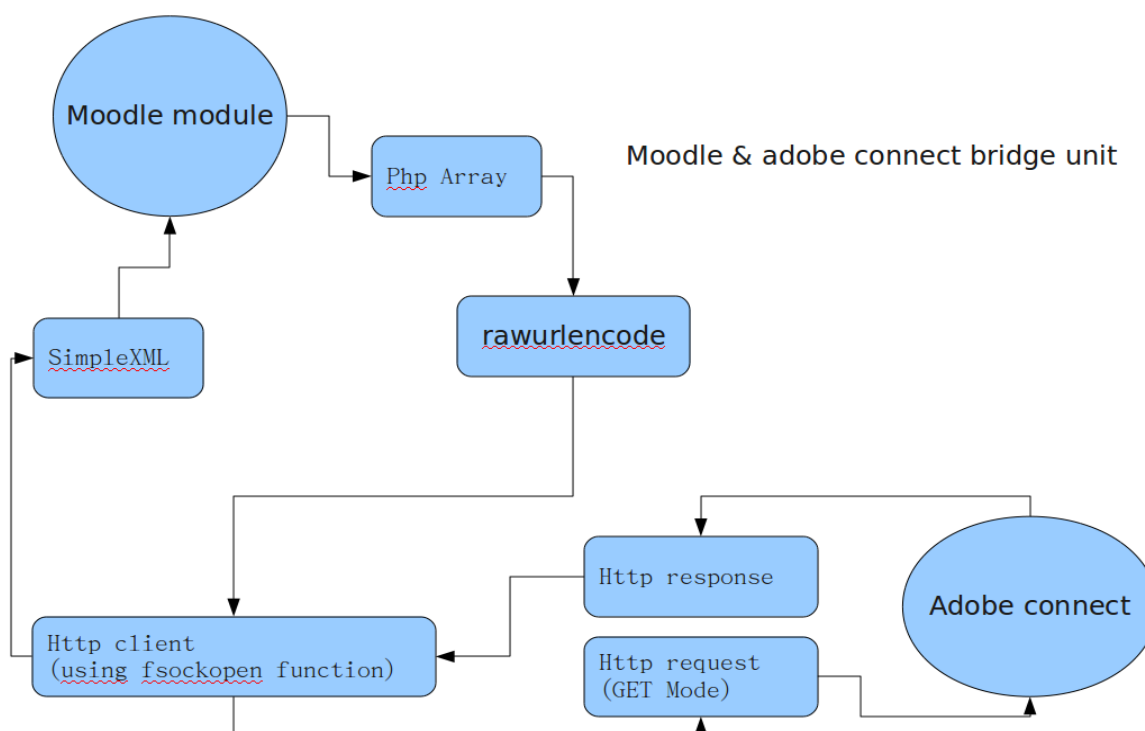


Fig. 3 Moodle & Adobe Acrobat Connect Pro bridge unit

In the bridge unit, the messaging process works according to the following steps:

1. Requests of every Moodle module were set into an array. This array can be encoded to a http request by using “rawurlencode” function and then using “fsockopen” to create http request for the Adobe Acrobat Connect Pro to play send and request functions.
2. Whenever getting the response from Adobe Acrobat Connect Pro, SimpleXML would be used to parse this response. A return SimpleXMLElement class would be send to Moodle by SimpleXML module. Moodle module can use this class to get useful data.

3.2 user correspondence

There are many roles in Moodle, and every role has different permission and rules. In Adobe Acrobat Connect Pro, there are also serial principals and every principal has different permissions on different scope. It is needed to define the correspondence between Moodle roles to Adobe Acrobat Connect Pro principals.

Table 11 Moodle roles& Adobe Acrobat Connect Pro Permission

Permission	Roles
manager	Admins
publish,host	Teachers
view	Students

When user try to create a new meeting activity on a Moodle platform, Moodle will find out who are teachers and students of this course for checking the status of the principal:

1. If there is no existed user belong to this course, module will create a new one
2. If his Principal is incorrect, this module will fix it.

3.3 synchronous learning activity module

Basic process for establishing synchronous communication was defined according to each role of Moodle course users. Synchronous learning activity

module could be design based upon the following sequence defined for each role.

1. teacher:
 - 1 add new instance of modules in course
 - 2 set up or update an instance
 - 3 list all instances in a course
 - 4 view a particular instance
2. student:
 - 1 list all instances in a course
 - 2 view a particular instance
3. administrator of Moodle:
 - 1 install module
 - 2 defines the structure of db tables for all database types
 - 3 backup & restore data
 - 4 setup global value for all instance of module

When teacher want to assign a on-line learning activity, the class of “widget_add_instance()” would be applied. The “Create the URL to a meeting room for which the user is host” process could be used to create a meeting on Adobe Acrobat Connect Pro server, and then its sco-id would be send to Moodle. Finally, this sco-id will store in Moodle database. When teacher want to launch a synchronous learning activity module:

1. Once the user is authenticated on Moodle, log the user in to Connect Enterprise
2. Get the value of the BREEZESSION cookie for the user’s session.
3. In the Moodle module, a meeting room URL is create
4. Open the meeting room URL with the guest name parameter.

3.4 Coding Structure between Platforms

Coding structures of both platforms of Moodle and Adobe Acrobat Connect Pro were identified the relationship for the purpose of implementing synchronous learning activity. Files and functions of Moodle platform were identified to the corresponding Adobe Acrobat Connect Pro Principal and Meetings Process

All those necessary corresponding pairs build a module were pointed out. In Table 12, for example, when Moodle call index.php script, the index.php maybe use “Log in using the session parameter” or “List all meetings on the server” in Adobe Acrobat

Connect

Pro

Table 12 module files& Adobe Acrobat Connect Pro function

Moodle file	Moodle function	Adobe Acrobat Connect Pro function
index.php		Log in using the session parameter
		List all meetings on the server
		Find the sco-id of a meeting
view.php		Create the URL to a meeting room for which the user is host
		Create the URL to a meeting room for which the user is not host
		Launch a meeting and let the user enter as guest
lib.php	_add_instance()	Create a private meeting and add host, presenter, and participants
		Add a user to a meeting and display meetings
	_update_instance()	Create a private meeting and add host, presenter, and participants
	_delete_instance()	Remove a user’s permission to access a meeting

4 Conclusion

A graphical user interface is provided for Creating a Synchronous Activity, as shown in Fig. 4. Teacher

can select which Adobe Acrobat Connect Pro server would be used for this activity.

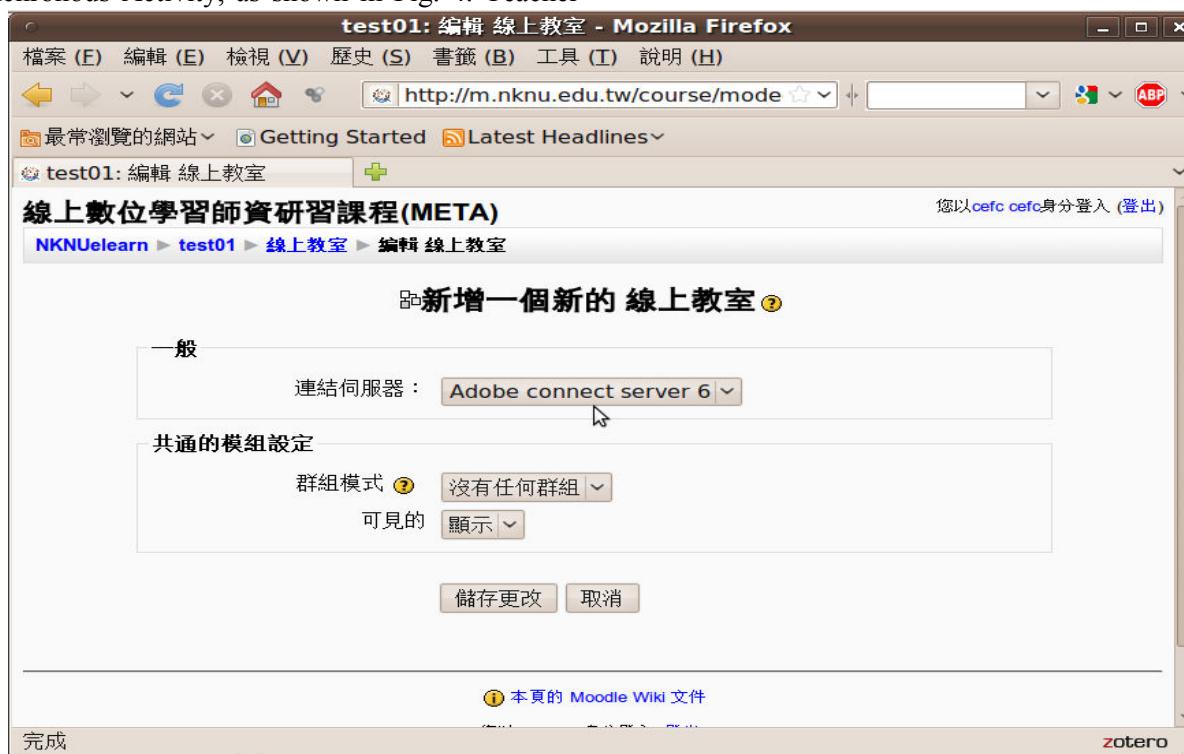


Fig. 4 Graphical user interface for Creating a Synchronous Activity

When teacher created a Synchronous Activity in a course, teacher and students can view at index page, Fig. 5, and then they can check this link to show Synchronous Activity view page, Fig. 6. Teacher and

students can see not only a link for accessing Adobe Acrobat Connect Pro meeting, Fig. 7, but also records of Adobe Acrobat Connect Pro meeting.

Fig. 5 index page of Moodle course





Fig. 6 view page of Synchronous Activity

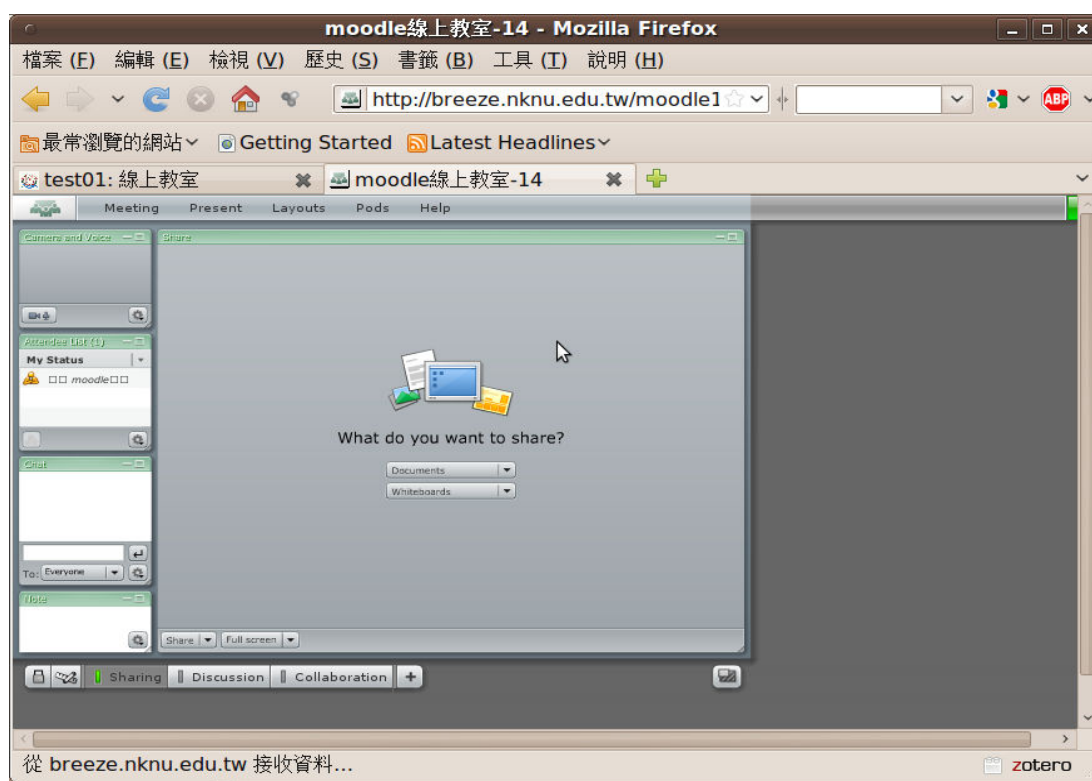


Fig. 7 Adobe Acrobat Connect Pro meeting

This study proposes a Synchronous Activity Module for the Moodle CMS. Based on the proposed Module, Moodle really will use for full online classes as well as supplementing “face-to-face” learning and help students to improve their learning progress.

References:

- [1] G. Gorghiu, M. B[^]zoi, L.M. Gorghiu, and A.M. Suduc, “Aspects related to the usefulness of a

distance training course having Moodle as course management system support,” *Proceedings of the 9th WSEAS international conference on Distance learning and web engineering*, 2009, page. 54–59.

- [2] G. Liodakis, M. Kalogiannakis, M. Psarros, 及 K. Vassilakis, “

Building E-services for Learning and Teaching by the Exploitation of an LMS System,” *Proceedings of the 9th WSEAS International Conference on Communications table of contents*, 2005.

[3] <http://www.cmsreview.com/>
[Online] Available:

<http://www.cmsreview.com/>

[4] Moodle - A Free, Open Source Course Management System for Online Learning. [Online]

Available: [_http://moodle.org](http://moodle.org).

[5] Features - MoodleDocs. [Online] Available:

http://docs.moodle.org/en/Features#Course_management.

[6] S. Pastore, “Social networks, collaboration and groupware software for the scientific research process in the web 2.0 world,” *Proceedings of the 7th WSEAS International Conference on Artificial intelligence, knowledge engineering and data bases*, 2008, page. 403–408.

[7] AdobeR AcrobatR Connect? Pro for eLearning. [Online] Available: http://www.avoka.com/ad_connect/resources/eLearningSolutionBrief.pdf.

[8] Adobe Acrobat Connect Pro guide [Online] Available: http://www.adobe.com/education/resources/hed/instructional/connect/collaborative_teaching/pdfs/overview_connect.pdf.

[9] Development:Modules - MoodleDocs. [Online] Available:<http://docs.moodle.org/en/Development:Modules>

[10] Development:Roles - MoodleDocs. [Online] Available:<http://docs.moodle.org/en/Development:Roles>

[11] Using Adobe Acrobat Connect Pro 7 Web Services. [Online] Available:http://help.adobe.com/en_US/AcrobatConnectPro/7.0/connectpro_7_webservices.pdf