Information for continued development of the moodle mod/plugin

**Document Created 28th August**

**Author: Samuel Deane**

Contents

[Ongoing questions 1](#_Toc395624278)

[Problems 1](#_Toc395624279)

[To-Do/Implementation List 2](#_Toc395624280)

[Purpose 2](#_Toc395624281)

[Background information 2](#_Toc395624282)

[Literature Review 2](#_Toc395624283)

[Setting up moodle in windows 2](#_Toc395624284)

[Testing of the Virtual Programming Lab Module 3](#_Toc395624285)

[Project focus 4](#_Toc395624286)

[Current Code layout of the mod 4](#_Toc395624287)

[Testing the current code and operations 5](#_Toc395624288)

[Code to extend the GUI / alterations to the code 5](#_Toc395624289)

[Netbeans & eclipse plugin 6](#_Toc395624290)

[General Notes 6](#_Toc395624291)

[Meetings (weekly starting 28th August 2014) 7](#_Toc395624292)

[References 7](#_Toc395624293)

[Appendix 1: PHP and Moodle scripting 7](#_Toc395624294)

[Glossary 7](#_Toc395624295)

# Ongoing questions

* What is the execution structure of a moodle module once installed
* How do you see the module working i.e. write a function/method to do …

Or write a class/classes to do … with a pre-defined API so that it can be tested?

* What level of feedback, form the code does not run to syntax errors etc. indicated by the compiler … run time errors or looking for known mistakes and adding helpful hints
* Should the activities tie with other materials

i.e. in the activity details section:

This activity is based on material covered in lecture 6, if you are having difficulty please read the Lecture Notes (Lecture 6 link), rewatch the lecture (Lecture 6 recording) or read chapter 3 pages 32-48 in the book

## For Paul

* I need some sample questions to
  + test the program
  + improve the layout

## For Net Spot

* is it possible to run a sandbox compiler service on the moodle machine?
  + Or using a separate dedicated sandbox service on the university server?
* Is the dual plugin method a good way of doing implementing the addon?
* Should I use classes over stdclass?
* Since JavaScript may not be enabled what is the best way to handle action events such as
* How to make a button do something
* How to make a select list work on change

# Problems

* ~~‘String does not exist. Please check your string definition for pluginname/assignfeedback\_assignfeedback’ appears on the settings page?~~ FIXED
* Checkpoint and test files have required fields but only if checkpoints and test are enabled (not special file only) … but even if these fields are aren’t enabled the rule still applies which causes a problem on the get settings page
* at the moment the checkpoints ordering (or checkpoint number) cannot be change i.e. reordering of checkpoints for a activity is not supported yet
* Assume that guard blocks (greyed out areas) will be transmitted as a list of lines
* Submission time remaining is calculated wrong
* The test function does not account for compiler return values (revals) other than 0 (compile successful) though may output it
* The outputerr can be run in compilation with command parameters to ascertain program exit values or compile return values (retval)s irrelevant

# Short term dev list

* Implement simple add checkpoint and task from menu
  + The gui
  + The saving of the data
  + The external web services
  + Make sure web service external library can handle it
* Implement compile with results in the user view
* Implement grading of assignments

# Long term dev List

* Finish the java port of the client
* Test the functionality of the client and web service
* Add a checkpoint add page in moode
* Add error to say moodle is not running for the client
* Submission of code through the Moodle web interface.
* Integration with the Moodle gradebook API into the codehandin module. This is nearly finished and will allow the topic teacher to export grades for all assignments in a topic.
* Online definition of checkpoints and tests using files. This should also allow checkpoint and test details to be updated/ deleted.
* In the grading interface, allow the assessor to run the tests again and see the output. This could probably be implemented using an AJAX call to the existing webservice.
* Improve sandboxing. This is implemented in the cmdRun function of externallib.php. Sydbox is a good candidate, but I have been unable to build and use it successfully. Ideally the sandboxing would not require compiling or installing extra tools, but use those tools that are built into the Linux system.
* Provide a webservice function that returns non-assessment test data for a given checkpoint. This would be useful for use in conjunction with an IDE or extension to the handin script, as it would allow the user to run the tests locally and use their own debugging tools if they wish, and would encourage students to develop in a test-driven manner.
* Improve the handin.py script so that the user isn't required to enter their username and password every time they wish to fetch their assignments, submit their code for testing and submission. This could be implemented by storing the users webservice token in a file in their home directory. If an IDE plugin were written that interfaced with the webservice, the token could be stored in a password store.
* Supporting group work for the Codehandin activity type, where multiple students work together and submit a single piece of work, and receive a single grade among them.
* Activity grades can either auto submit or require lecturer checking before final submission
* Integrate test with j/c/p etc unit testing perhaps
* Add a activity sub type where some code is already filled in i.e. more directed coding:

i.e. for java, the teacher would a uncompleted block of source code for a submission

Public class addN{

int a=4;

private int addNumber(int b){

/\* enterCodeHere \*/

}

}

The student would see

Public class addN{

int a=4;

private int addNumber(int b){

/\* enterCodeHere \*/ <- removable

return (a+b); <- is able to add code at this point only

}

}

* Develop some form of more detailed feedback (may be difficult as it would need to know many of the pitfalls that students might get themselves into.
* Produce a netbeans/eclipse plugin that has the same functionality as moodle but can compile and test on the students machine rather than going through moodle.

# Problems

* compile servers are hard coded
* limited compile process management

# Purpose

This document is intended to be a record of continued development of the moodle project

# Background information

## Literature Review

Read mine and johns poster and thesis’s + the majority of papers he mentioned

## Setting up moodle in windows

Moodle installed using the windows package (includes XAMPP (apache and PHP))

Might have to enable the PHP extension in the config file of XAMPP.

## Testing of the Virtual Programming Lab Module

A general guide can be found at: <http://cs.smith.edu/dftwiki/index.php/Moodle_VPL_Server_Setup>

Added vpl to the server\moodle\mod folder

Using cgywin (with gcc, libiconv, cygwin32-openssl, openssl and openssl-devel packages) to install the server

Commented out setrlimit(RLIMIT\_NPROC,&limit); in jail.cpp

Found that I was starting to port the jail server (changing to windows resource manipulation) so I installed moodle and vpl in ubuntu running through virtualbox

As per docs.moodle.org/26/en/Step-by-step\_Installation\_Guide\_for\_Ubuntu

Also note that will either place the moodle files directly into /var/www/html or

Change DocumentRoot /var/www/html to /var/www/html/moodle in

/etc/apache2/sites-available/000-default.conf

May have to also run #sudo apt-get install libapache2-mod-php5

To get apache to regconise php

Backed up the Ubuntu virtual HD (made a copy) of the clean moodle install before installing the mod and jail server

Installed the mod and jail server which installed several compilers and related packages (unit testers, automation testing etc.). However I am unable to get the jail server to work. Not sure about the terminology used

Some info that may be useful (but probably isn’t):

I suppose that you are installing a 2.0.1 jail server. You're using incorrectly the URLPATH, this variable is part of the [URL](https://moodle.org/mod/glossary/showentry.php?eid=31&displayformat=dictionary) not the full URL. For example, if you set URLPATH=/HELLO your URL for your service must be <http://jailserver.it:8080/HELLO>.  
The message you found say that your PHP don´t has the xmlrpc extension. VPL requires this PHP extension to communicate with the jail server.

# Setting up the Development Environment

Prominently netbeans was used as the development environment.

This included installing a Python module to develop the client, though it is assumed that an equivalent java based version will be developed to integrate into the netbeans and Eclipse IDE natively.

Python 2.7 had already been installed as part of setting up for supported programming languages of the plugin. However to get the client to work the python package ‘poster’ was also installed.

On windows this involved opening a command prompt in the python installations Scripts folder

i.e. cd c:/Python27/Scripts and then >easy\_Install.exe poster (or could use the .py versions)

This allowed the

from poster.encode import multipart\_encode

from poster.streaminghttp import register\_openers

import statements to run.

NOTE: that while the upgrading of the client code to python 3.4 (or the most recent python version) does not present a problem (predominantly syntax changes – print “a”,b -> print (“a”,b) and similar for exceptions), this would also require an upgrading of segments of the poster package (multipart\_encode, register\_openers and their dependancies) which may be more involved without much return at this point.

To make python easier to use it was added to the windows path

Control panel -> system -> advanced System Settings -> Advanced tab -> Environmental variables

Under system variables, Path click edit and add C:\Python27 or wherever python is

# Project focus

In two parts …

## Current Code layout of the mod

See my thesis

## How the plugin works

Uploads contain two parts, a zip file containing uploaded files and a series of

# Testing the current code and operations

PHPUnit provides the current testing framework that is supported be moodle

This can be implemented through the Netbeans IDE

<http://docs.moodle.org/dev/PHPUnit>

* This details information about using custom PHP scripts

… I haven’t done much with this

… to test create a folder called ‘tests’ to the project directory and create files of the same name as the file to be tested and appending ‘\_test’ to each.

# Upgrades to the previous project

**Replacement of deprecated calls …**

get\_context\_instance() deprecated since, use context\_xxxx::instance() instead

i.e. $context\_system = context\_system::instance();

or $context = context\_user::instance($USER->id); as per <https://docs.moodle.org/dev/Access_API>

add\_to\_log() has been deprecated, please rewrite your code to the new events API

see http://docs.moodle.org/dev/Migrating\_logging\_calls\_in\_plugins

some of the sql queries did not work (aid and .ordering). change to .id in both cases … not sure if this will lead to errors down the line …will correct as needed

formslib.php not found when adding an activity

still not to sure what the error relates to after reading <http://docs.moodle.org/dev/lib/formslib.php> and <http://docs.moodle.org/dev/lib/formslib.php_Usage>

**rewrote some of the xml db schema** as per <https://docs.moodle.org/dev/XMLDB_editing> to

* include foreign keys
* added the ‘ordering’ field to codehandin\_checkpoint as it was used throughout the code but did not exist within the schema.
* codehandin submission changed id to assign\_id
* changed the naming convention used for ids such that a\_id becomes aid as per the rest of the database.

Added a settings / config file <https://docs.moodle.org/dev/Blocks#Basic_Concepts>

//a file picker for single file upload

$mform->addElement('filepicker', 'userfile', get\_string('file'), null,

array('maxbytes' => $maxbytes, 'accepted\_types' => '\*'));

// a file manager for multi file upload

$mform->addElement('filemanager', 'attachments', get\_string('attachment', 'moodle'), null,

array('subdirs' => 0, 'maxbytes' => $maxbytes, 'maxfiles' => 50,

'accepted\_types' => array('document') ));

// an editor field

https://docs.moodle.org/dev/lib/formslib.php\_Form\_Definition#editor

# Netbeans & eclipse plugins

Layout



1 2 3 4 5

added an extra combo box (like 2) for the course name and left 2 as the activity

Legend:

1. Settings … set the user name id, login etc.

1.5 the course name (not added to the figure)

1. Set the test suite/activity to use
2. Download / update (if worked on through moodle) the activity
3. Upload/submit the activity to moodle
4. Test the current code against the activity/suite

Features

* Establish a connection with moodle
* Download, upload/submit code to moodle
* Test the code using net beans inbuilt compilers and testing interface

Netbeans Coding

<http://wiki.netbeans.org/DevFaqOutputWindow>

* Writing to the output window

Project handling

Create project - ????

... get the template … put in "TemplateWizard"? … use wizard to create project

Open project - http://wiki.netbeans.org/DevFaqOpenProjectProgramatically

Get project of active file - <http://wiki.netbeans.org/DevFaqGetProjectForFileInEditor>

Guardblock use:

It is assumed that some sections of the code will be non-editable by the students. In netbeans this can be achieved through the use of guard blocks, an exert on guard blocks from net beans follows:

### Using guard blocks - <http://bits.netbeans.org/dev/javadoc/org-openide-text/org/openide/text/doc-files/api.html>

Manipulating guarded blocks should not be very difficult, if your application requires it. They are used by the FormEditor, but any module which needs to prevent the user from editing certain portions of a text document (typically because they will be mechanically recreated by other means), can do so.

* To create a guard block over an existing area of text, you may use [NbDocument.markGuarded(...)](http://bits.netbeans.org/dev/javadoc/org-openide-text/org/openide/text/NbDocument.html#markGuarded%28javax.swing.text.StyledDocument,%20int,%20int%29); the guard block may subsequently be removed using [NbDocument.unmarkGuarded(...)](http://bits.netbeans.org/dev/javadoc/org-openide-text/org/openide/text/NbDocument.html#unmarkGuarded%28javax.swing.text.StyledDocument,%20int,%20int%29).

Typically you will want to remember the positions of the guard blocks you added using a position, so that user edits in the vicinity of the guard block will be taken into consideration. You may create such a position using [NbDocument.createPosition(...)](http://bits.netbeans.org/dev/javadoc/org-openide-text/org/openide/text/NbDocument.html#createPosition%28javax.swing.text.Document,%20int,%20javax.swing.text.Position.Bias%29), and retrieve its current offset when needed using [Position.getOffset()](http://download.oracle.com/javase/7/docs/api/javax/swing/text/Position.html#getOffset%28%29).

* To insert a new guarded block of text, you may use [NbDocument.insertGuarded(...)](http://bits.netbeans.org/dev/javadoc/org-openide-text/org/openide/text/NbDocument.html#insertGuarded%28javax.swing.text.StyledDocument,%20int,%20java.lang.String%29).
* To change the contents of part of the document, without regard to the presence of guard blocks (this assumes you know what you are doing and what the guard blocks are being used for), you may use the standard Swing [Document.insertString(...)](http://download.oracle.com/javase/7/docs/api/javax/swing/text/Document.html#insertString%28int,%20java.lang.String,%20javax.swing.text.AttributeSet%29) and [Document.remove(...)](http://download.oracle.com/javase/7/docs/api/javax/swing/text/Document.html#remove%28int,%20int%29). You probably want to use [NbDocument.runAtomic(...)](http://bits.netbeans.org/dev/javadoc/org-openide-text/org/openide/text/NbDocument.html#runAtomic%28javax.swing.text.StyledDocument,%20java.lang.Runnable%29) to prevent errors in threaded code.

To do the same sort of thing while preventing yourself from accidentally touching a guard block, i.e. if your module was not the creator of the guard block (or you are not even sure if there any in the document), please use [NbDocument.runAtomicAsUser(...)](http://bits.netbeans.org/dev/javadoc/org-openide-text/org/openide/text/NbDocument.html#runAtomicAsUser%28javax.swing.text.StyledDocument,%20java.lang.Runnable%29) instead.

# General Notes

Moodle info

* Apache 2.4.4
* PHP 5.4.25 (VC9 X86 32bit thread safe) + PEAR
* MySQL 5.5.32 (Community Server)

The package also includes **Moodle 2.7.2+ (Build: 20140911) (2014051202.01)**.

On the Administration pane under Site Administration/advanced features and set Enable web services to true

To add a course go to courses and category management click on a category (for example Miscellaneous), then Create new course (near the bottom of the page).

Under enrolled user click the silhouette of a person with a (+) symbol on the right side of the roles column.

Under Manage protocols make sure the REST eye is not striked out

CFG pointers can be located in moodle\config.php and moodle\install.php

# Meetings (weekly starting 28th August 2014)

See my thesis for a project description and meeting information

# References

<http://pdf.aminer.org/000/369/573/virtual_programming_lab_for_online_distance_learning.pdf> - A good reference for what e-learning is all about

# Appendix 1: PHP and Moodle scripting

### PHP general

Use $this to refer to the current object. Use self to refer to the current class. In other words, use $this->member for non-static members, use self::$member for static members.

### Moodle

lib/moodlelib.php is where the PARAM\_ types are defined

Any place where a user can input content that is output by format\_text, $options->noclean, must be protected by a capability check, and the capability must be marked as **RISK\_XSS**.

* <http://docs.moodle.org/dev/Security:Cross-site_scripting>

<http://docs.moodle.org/dev/Data_manipulation_API> - how to manipulate data in the database

$plugin->cron=0 says that will run every cron job (for most servers this is every 10 minutes)

stdClass is PHP's generic empty class, kind of like Object in Java or object in Python

foreach (array\_expression as $value)

statement

foreach (array\_expression as $key => $value)

statement

Scope Resolution Operator (::), allows access to static, constant, and overridden properties or methods of a class.

‘=>’ assigning a value to a named key

# Other notes/questions

* How do you install the vpl jail server?
  + What is the serer name?
  + Is the ‘server’ the same as the execution server?
  + How do I run it through the local host?
  + Is it possible to have more than one thing running through the local host
  + Do ports mean I can have multiple?
* Where do I set the programming language for a vpl activity

# Glossary

Free and Open Source Software FOSS

# Project detail

## File System organisation

There are two ways to organise files on the file system

1. Hold an entire assignment in one filearea and use the assignment id to specify it. Then to access an individual test use the test id and type as a pathname

i.e under filearea = ‘chiAssignments’ (filearea for all assignments)

where itemid = assignmentid files are arranged like

testid/[type]/file

or [type]testid]/file

1. Hold each test file in one of three fileareas by its type (i.e one for input and one for output and one for output error (outputerr)) and use the assignment id to specify it

i.e under filearea = ‘chiAssignmentInput’ (filearea for input of all assignments)

where itemid = testid files are arranged like

inputfile (there is only one file)

similarly under filearea = ‘chiAssignmentOutput’ where id = testid

outputfile (there is only one file)

likewise under filearea = ‘chiAssignmentOutputErr’ where id = testid

outputerrfile (there is only one file)

Method 1:

Advantages: all the files for a single assignment are held together making the zipping of an assignment far easier

Disadvantages: Since draft fileareas mirror real fileareas there are no functions to handle placing a file from a real filearea that contains multiple files into its own draft filearea. Also files must be extracted and inserted into the same file area making separate updates difficult.

Method 2:

Advantages: files a spread out making them far easier to access individually

Disadvantages: no function exists to get the files from multiple fileareas.

If there is no browser based file modification allowed then the choice is the first one as no additional functions or file modifications are required whereas the second method requires a new function: get\_multiple\_area\_files that allows multiple files to be taken from multiple itemids. However if browser based modification is required the main disadvantage of the first method comes into play, that is, getting a single file from within a draft area and placing it in its own draft area. To achieve this two methods must be modified - file\_prepare\_draft\_area and file\_save\_draft\_area\_files of filelib.php

Files are stored in one file area 'submission\_chi'

With component name 'assignsubmission\_codehandin'

With files paths ike so

/g/cpid/testid/o/file.ftype

/t/cpid/testid/o/file.ftype

*// Read contents*  
if ($file) {  
 $contents = $file->get\_content();  
} else {  
 *// file doesn't exist - do something*  
}

Lib\ddl\mysql\_sql\_generator.php

See getTypeSQL for type conversion

TYPE="int" type depends on length -

if ($xmldb\_length > 9) {

$dbtype = 'BIGINT';

} else if ($xmldb\_length > 6) {

$dbtype = 'INT';

} else if ($xmldb\_length > 4) {

$dbtype = 'MEDIUMINT';

} else if ($xmldb\_length > 2) {

$dbtype = 'SMALLINT';

} else {

$dbtype = 'TINYINT';

}

<http://www.java2s.com/Code/Jar/j/jersey-bundle.htm>

token,php (around line 148) needs to updated so that tokens can be generated on the fly for this web service otherwise additional tokens will need to be created

### How the file system for the mod works

The system only uploads changes back to the moodle server to be processed. As such all data sent to moodle is assumed to have changed and will overwrite old data even if they are the same

1. The test files are uploaded in a zip under the following file paths

/[g|t]/[n[ordering]|checkpointid]/ [n[ordering]|testid]/[i|o|e]/[filename]

Orderings and checkpoints ids in regex are \d{1,10}

Or in words:

g (grade only) or t (for test and grade)/

checkpointid if the checkpoint exists or n followed by the ordering number if it does not/

testid if the test exists or n followed by the test ordering number if it does not/

the type of file I (for input), o (for output) or e (for output error)/

the filename

Note: The ordering number that defines the order in which test are run (lower number first)).

Examples:

* + i.e. for an existing grade only (==g) test of checkpointid 5, testid 4 and type input (i) with a filename of in.c would be stored under the path: /g/5/4/i/in.c
  + i.e for a new submission and grade test or a non-grade only (==t) test of checkpointid 6 and ordering 8, testid 4 and ordering 1 and type input (o) with a filename of out.c would be stored under the path: /t/n8/n4/o/out.c

1. The new data is then inserted/updated in the database
2. The zip file is then extracted into the moodle file pool under the assignmentid as the itemid, ‘assignsubmission\_codehnadin’ as the filearea replacing any [n[ordering] ids of new checkpoints and test with their actual checkpoint and test ids respectively returned by the previous insertion of the checkpoint and test data.
3. Two zip files are then created one that contains student testing files (non-grade only) - tassignmentid.zip and another that contains all the test files for marking and updating of the codehandin instance - gassignmentid.zip

The tassignmentid.zip file is created by compressing all files under the /t directory (in the moodle file storage pool) and the gassignmentid.zip files is created by compressing all the files under the /g/ or /t/ root folders (in the moodle file storage pool). These files are created directly into the codehandin folder in the defined moodle data folder so they can be downloaded later by users.

1. The address for these zip files can then be created using the information supplied by fetch assignments function or specified using the fetch assignment files url function. Both functions rely on the plugin.php code to download files
2. The zip fie is then download and extracted into the predefined assignments file folder under /assignmentid/a/ …. File path ( as specified in 1 with no /t or /g part)
3. The files can then be read through the client to see what the input is and the expected outputs are

A sample java client has been created

add\_plugin\_submission\_elements($submission,

MoodleQuickForm $mform,

stdClass $data,

$userid) {