IMPROVED ASSIGNMENTS MANAGEMENT IN MOODLE ENVIRONMENT

Katarína Jelemenská, Pavel Čičák
Slovak University of Technology Bratislava, Faculty of Informatics and Information Technologies, Bratislava (SLOVAKIA)
jelemenska@fiit.stuba.sk, cicak@fiit.stuba.sk

Abstract
The purpose of this contribution is to present a solution that was developed to support Digital Systems Description course. In this course the students should comprehend the methods and techniques used in digital system design, and gain the skills in digital systems modeling using hardware description languages (HDLs). The lab assignments in this course used to be managed and evaluated manually. In addition to tedious work it required, it brought about other issues like non-objective and non-realistic assignments evaluation or plagiarism attempts. Nowadays there is a common practice to use LCMS (Learning Content Management Systems) tools for course management, study materials creation, presentation, and management, as well as assignments and tests management. However, most of the LCMS support only the basic functions for assignments management like collection of students’ works with or without timing restrictions, manual grading and feedback providing. These functions do not fully satisfy the assignments management requirements of the course. That is why the new module was designed, suitable for more or less automated tasks to students allocation and solutions assessment in the area of digital systems modeling. Among other functions the module automatically verifies plagiarism attempts, as well as the syntactic correctness and simulation results of the solutions using external compiler and simulator. The paper presents our first experience with course assignments management redesign based on Moodle LCMS (Learning Content Management Systems) using the special Moodle extension recently developed at the Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava. The module provides full supports for VHDL assignments. Other HDLs are partially supported as well.

Keywords: assignments management, grading, automated evaluation, Moodle, VHDL.

1 INTRODUCTION
Web based training is currently a very popular method of education. It represents modern trends in application of new educational forms, based on convenient utilization of capabilities, provided by IT technologies. Various Content Management Systems (CMS), Learning Management Systems (LMS), Computer-Based Training (CBT) or Web-Based Training (WBT) systems can be used supporting various aspects of educational process. Nowadays there are a vast number of such systems, but most of them are unable to support automated knowledge and/or assignments assessment in the courses devoted to programming languages. Whether it was assembly language, high level programming language, or Hardware Description Language (HDL) the knowledge and skills assessment has always been an issue.

To master the programming or modeling language, the students should solve various assignments in the respective language. However, to correct and evaluate the programs or models manually by teachers is not an easy task. Unfortunately the choice of automated assignments assessment systems devoted to HDLs is very limited. Therefore we have years of experience with manual assignments assessment. The lab exercises in Digital system description course are based on more or less individually allocated assignments that the students can solve at school or at home. The solutions are evaluated by lab instructors and usually form a part of the overall course assessment. The quality assessment process should not only evaluate the knowledge and skills the students gained but also give them a useful feedback that can improve them. Unfortunately, there are always some students who try to find the easiest way to get the lab work results (e.g. search for a complete solution on the internet, copy the solutions from their classmates, etc.). In case the manual approach is adopted to assignments assessment the lab instructors usually have no chance to find out whether the students really did the programming work themselves, or not. They can just check up, whether the students understand their solutions and whether they are able to perform some minor changes in them. But it is
not an easy task and different teachers would not probably strive equally to be rigorous which might result in an inconsistent assignments evaluation. As a result, this part of the course assessment is often neither objective nor realistic although requiring a lot of teachers’ work. The situation is even worse when distant learning is considered, where teachers do not have the possibility to check out the students understanding at all.

Due to the increased number of students it is not possible to assign unique tasks to each student. Instead a restricted number of more or less similar tasks is available for each assignment that need to be distributed among the students in such a way that at least each student in the same group will be assigned a unique mix of tasks. This is also a time consuming and error-prone work when performed manually.

To make the students’ assignments management and evaluation easier and more effective a Learning Content Management System (LCMS) or at least some dedicated computer based tool should be used. Systems that automatically assess student programming assignments (requirements satisfaction, programming style, originality etc.) have been designed and used for over forty years [1], [2], [3], [4]. However, they are typically designed for one or several related programming languages. As long as C++ or Java languages are covered massively, we were not able to find any application devoted to HDLs. As far as LCMSs are concerned some of the currently available systems support simple assignments management like collection of students’ works with or without timing restrictions, manual grading and feedback providing (e.g. Moodle [5], ATutor [6], Ilias [7], and Toolbook [8]) but no automated evaluation. The above mentioned problems encouraged us to concentrate our work on new applications development that would enable sophisticated assignments management, testing of knowledge, understanding, and especially the gained skills, all in an automated manner. Several applications devoted to skills based testing have already been designed, implemented and integrated into the educational process at the Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava, most of them oriented toward the HDL design and other behavioral specification means [9], [10], [11], [12]. In this paper we will concentrate more on the assignments management issue presenting the special module for Moodle LCMS that will improve the Moodle assignments management capabilities.

2 SYSTEM REQUIREMENTS

The assignments management requirements in the course Digital system description provided at the Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava exceed widely the basic functions offered by currently available LCMSs. As the compulsory course it is attended by 50-100 students a year and the labs are fully based on the assignments. During the term each student is designing four digital system models in three HDLs – two VHDL models, one in SystemC and one in Handel-C. The help in a form of a system performing automated tasks distribution and solutions evaluation is therefore highly required. We identified following requirements that should be satisfied when designing such a system:

- the system should be web-based with permanent availability so that teacher and students can access it any time and from any place;
- the teacher should have the possibility to create and manage tasks that will be classified according to the assignments types;
- each assignment type should have a solution deadline defined together with penalty conditions for late solution delivery;
- there are several tasks variants for each assignment type that should be assigned to students so that each student in the same group will solve different tasks;
- the system should be able to collect and store identified solutions consisting of several files;
- the HDL source codes should be checked for syntactical and functional correctness and eventually for quality in terms of programming style which should result in solutions assessment;
- several sophisticated algorithms should be combined together to get the best possible results of automated solutions grading;
- the solutions should be checked for plagiarism attempts in the same group, across the groups and also within the previous school years archives;
- the automated evaluation should be combined with the solutions collection in a reliable way to prevent possible capacity or throughput issues;
These were the basic requirements that were followed during the advanced assignments management module development that will be presented in this contribution.

3 SYSTEM FUNCTIONS

The system functions were derived from the activities that have to be done to manage student’s assignments in a manner that will satisfy the basic requirements specified in the previous section. The system should recognize four types of users – teacher, lab instructor, student, and administrator. The teacher is responsible for the course and therefore for this role the highest permissions are required in terms of the course management, including the assignments management. The lab instructor will have the permissions to manage the tasks assignment and solutions evaluation of the groups of students he/she was scheduled to teach. A user with the role student can access just his/her own tasks, solutions, and assessments which includes the possibility to upload the solutions. For the last role, administrator, only the backup functions are needed.

In case the system will be integrated into the Moodle LCMS some of the functions like users and roles management are already implemented and can be shared by the new module. Thus only the functions related to the advanced assignments management have to be taken into account. The module functions are divided into three categories: assignments management, solutions management, and solutions assessment.

3.1 Assignments management

Assignments management consists of essential functions, available to teacher, which allow him/her to create, edit, delete, and distribute tasks to students. The feature is important especially in order to establish more extensive database of different tasks that will reduce the space for plagiarism in assignments solutions. The secondary function is to generate an entry for the solutions testing module that will identify and select the correct environment description for the task based on the association of the student and the task assigned to him/her.

3.1.1 Task creation

The function enables to create a new entry using the internal editor supporting basic text and graphics. The data necessary for the task creation are: the task name, the maximum number of points for the developed solution (divided between the code and documentation), and the late delivery penalty (in percentage or points). The internal editor will allow to upload images to the server and then to use the relative path to the image in the generated text (ensures consistency of the task). The output of the new task will follow the HTML syntax with links to the images routed directly to the server. Each created task has to be associated with the assignment category to which it belongs.

3.1.2 Task modification

This feature, as the name suggests, is used to modify the content of an edited entry and to perform its renewal in the database. For this purposes the internal editor is used as in the function new task creation. The function allows also to change the assignment category – i.e. in case of category renaming it is not necessary to create the task again. The function is only accessible to the teacher.

3.1.3 Task deletion

Deletes the task. This function has limited usage - it is not possible to delete the task that was assigned to at least one of the students. The reason for this restriction is to ensure the database integrity with respect to the archival part.

3.1.4 Assignment categories administration

The function enables to create a new assignment category. Only the assignment category name is required for this purpose. The feature includes also category modification and deletion functions. An example of teachers user interface for these functions is given in Fig.1. The screen displays four assignment categories named Zadanie1 – Zadanie4. For each assignment category the time period is specified within which the solutions can be submitted.
3.1.5 Tasks allocation

This function serves to facilitate the work of teacher to distribute tasks to students while ensuring the maximum possible impartiality of the allocation. The function is dependent on the configuration data in a given activity. During the allocation process the task list table is used containing the assignment category for each task. The core function is automatic allocation performing random association between the tasks from database and the students of the course taking their group affiliation into account. The automated allocation can also be done within the selected group. The feature allows manual tasks allocation to students as well, and in this mode provides a warning message in case the task has already been assigned to another student in the course or in the same group respectively. An example of teachers user interface used for tasks allocation is given in Fig. 2.

3.2 Solutions management

Solutions management represents the functions which are predominantly used by users with students role like list allocated tasks, upload task solution, delete task solution, test the solution, send solution for assessment, display assessment. The solutions management has to consider also the possible user issues that need to be treated before the permission to upload the solutions is granted, e.g. the acceptance criteria for students’ solutions. A task solution can be accepted within the given period of time (specified during the task creation) or after the deadline in which case the late delivery penalty has to be applied. An example of teachers user interface used for tasks allocation is given in Fig. 2.

3.2.1 Upload task solution

The function allows to send the task solution to the server for further analysis. The function is implemented in a simple form using HTML and AJAX technologies, which enables to upload several files in one step. The uploaded file will be processed and validated by additional functions called thereafter. The script will save the file to the server retaining its name in its original form and will provide additional features such as the size and file type restriction in order to overcome the system overloading caused by possible attacks of this kind.

3.2.2 Delete task solution

Function to remove the already uploaded (unneeded) solution from the server. The function will be useful in cases where a student finds out that the uploaded solution is not correct (e.g. it would not compile) and needs to be replaced by the newer version. Of course the function allows the student to manage only his/her own solutions that were not sent for assessment yet.
3.2.3 Test the solution

The student has an option to compile the solution directly on the server before sending it for assessment. The compilation protocol will be displayed and in case of problems the student can remove the current solution from the server and upload the revised version. The compilation has only an informative nature and does not result in solution assessment.

3.2.4 Send solution for assessment

The function passes the uploaded files to the system for the purpose of further assessment. As a result the automatic scripts are called to verify the syntactical correctness of the solution and to check the solution for similarities to other solutions of the same task. The solution is also available for assessment by teacher or lab instructor. The need for teachers intervention in the assessment process is still necessary since not all the steps can be automated. A typical example might be the assignment documentation.
3.3 Solutions assessment

Evaluation of the solution correctness is based on the approach published in [13]. For this purpose the components of the commercial software ModelSim from Mentor Graphics Company are used that are installed on the server and controlled by shell scripts. The VCOM compiler is used to check the syntax and to compile the solution. The compiler output is simple text summarizing the analysis and compilation progress. For simulation purpose the VSIM simulator is used generating the simulation results in the form of a file with extension .wlf, in which the signals progress is contained. The VSIM tool allows the comparison of two .wlf files (tested and referenced) which can be used to assess the solution functionality.

The second part of the solution evaluation is the plagiarism detection. The approach used for this purpose is based on the YAP3 (Yet Another Plague) method [14] and RKR-GST (Running Karp-Rabin Greedy String Tiling) algorithm [15] since this method provides the best results compared to other considered methods and does not have enormous hardware requirements which is important when large number of solutions has to be processed. The lexical analysis and tokenization of input VHDL files is performed by shell scripts called from PHP environment and YAP3 method is implemented in C++ programming language.

3.3.1 Functionality assessment

The system allows to evaluate the syntactical correctness of the solution as well as the solution simulation using the predefined testbench for the given task. In case of syntax errors that make the simulation impossible, the solution is assigned zero points. The function can be activated manually and it is accessible to teacher and lab instructor. The primary function is simulation of all the solutions and their assessment based on the comparison of simulated signals. It can be used also in case of additional evaluation adjustments.

3.3.2 Automated plagiarism detection

Function accessible to teacher and lab instructor. Also in this case the function is triggered after solution was sent for assessment. The function checks the solution similarities to other solutions of the same the task. In this way the verification of all the solutions will be ensured. The function can also be activated manually in which case the solutions for comparison can be chosen. The feature is implemented by means of shell script call using the comparative method YAP3 with the parameters determining the path to the solutions to be compared (the output of the SQL query from the database). The script provides the degree of similarity as the return value and this value will be entered into the output table.

3.3.3 Manual plagiarism detection

Function accessible to teacher and lab instructor. This function is not necessary, however, it can be used to treat the unforeseen situations, when the teacher might want to compare the specific solutions. The function can be configured and any solutions can be specified. When suspected, it is thus possible to compare the solutions that do not belong to the same task.

The plagiarism detection results are displayed in an output table describing the relationships between the works and their similarities. The full form output (round robin) could be difficult to read when a large number of solutions is compared, therefore we have implemented two ways of displaying the solutions similarities. Primarily a listing of all the solutions in the form of a table is displayed, but only for one course and identical tasks. To ease the navigation the data in the table is sorted in descending order according to the level of similarity. The displayed data are: students names, submission time, course, and similarity level. An example of an output table is given in Fig. 4.

4 RESULTS AND EXPERIENCE

The designed and implemented Moodle module for advanced assignments management was tested for functional correctness and time demands. The functional testing revealed some issues mainly concerned with insufficient input variables verification that have been corrected continuously. The time demands testing involves all the components and their sum represents the overall time demands of the implemented module. This measurement was not supposed to offer exact set of results, but rather an indicative overview of the time demands of the individual module parts providing thus an area for efficiency improvements. The measurements were performed on the system with the following parameters: Intel Quad core Q6600 processor, 4 GB DDR2 800Mgz memory, Gigabyte EP45-UD3LR
motherboard, and Seagate 1TB, SATA hard disk. The PHP scripts were tested using the `microtime()` function directly in PHP code. The measured values were about several milliseconds, which represent negligible values that can not cause any risk.

The UNIX shell scripts were tested using the UNIX command "time". The shell scripts for lexical analysis and syntax or simulation verification were tested on two VHDL files - the "hard" one with 2144 lines, and the "easy" one with 113 lines of code. The results for one, ten and fifty VHDL files analyzed concurrently are summarized in Tab. 1. and represent statistical averages of 10 measured attempts. From the measured values an indication of the script difficulty can be considered for "hard" and "easy" solutions. The values show linear relation.

<table>
<thead>
<tr>
<th></th>
<th>&quot;hard&quot;</th>
<th>&quot;easy&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>real</td>
<td>user</td>
</tr>
<tr>
<td></td>
<td>user</td>
<td>sys</td>
</tr>
<tr>
<td>1 file analysis</td>
<td>0m0.243s</td>
<td>0m0.190s</td>
</tr>
<tr>
<td></td>
<td>user</td>
<td>sys</td>
</tr>
<tr>
<td></td>
<td>0m0.082s</td>
<td>0m0.040s</td>
</tr>
<tr>
<td>10 files analysis</td>
<td>0m2.334s</td>
<td>0m2.110s</td>
</tr>
<tr>
<td></td>
<td>user</td>
<td>sys</td>
</tr>
<tr>
<td></td>
<td>0m0.804s</td>
<td>0m0.520s</td>
</tr>
<tr>
<td>50 files analysis</td>
<td>0m11.708s</td>
<td>0m10.480s</td>
</tr>
<tr>
<td></td>
<td>user</td>
<td>sys</td>
</tr>
<tr>
<td></td>
<td>0m3.959s</td>
<td>0m2.670s</td>
</tr>
</tbody>
</table>

Similar tests were performed for the RKR_GST algorithm, which compares two previously processed files. The chosen methodology was the same as in the previous case. The results summarized in Tab. 2 display the measured time for file sizes 200, 500, 1000 marks respectively analyzing one, ten and hundred files concurrently. All the results show relatively small and acceptable delays.
### 5 CONCLUSIONS AND FUTURE WORK

In this paper the advanced assignments management module was presented developed for the Digital Systems Description course. The module can be easily incorporated into the LCMS Moodle and allows teacher to create and manage database of assignments divided into various categories, to assign them to students based on their group affiliation, and to archive all submissions and assignments created in the system. The module automatically verifies plagiarism attempts, syntactical and functional correctness of the submitted solutions which results in automated solutions assessment. Currently, the automated grading function is available only for assignments in VHDL but the rest of the functions supports other HDLs as well. The advanced assignments management module is now ready to be used in the course. The first experience shows that such an assignments management module can substantially ease the teachers work and save a lot of time in managing and grading process.

Currently the automated solutions assessment is based on the results of syntax verification and simulation results comparison against the model solution and does not take into account the quality of programming style or code efficiency. In this area we can see the space for further improvements. On the one hand, several sophisticated algorithms could be combined together to get the best possible results of automated solutions grading, on the other hand automated evaluation can be extended to other HDLs as well.

### ACKNOWLEDGEMENT

This work was partially supported by Slovak Cultural and Educational Grant Agency (KEGA 022STU-4/2012 "MEDU - technical courses mobile laboratory") and the Slovak Science Grant Agency (VEGA 1/1008/12 “Optimization of low-power design of digital and mixed integrated systems”).

### REFERENCES


