

GRAPHICAL AND FUNCTIONAL REDESIGN OF EXISTING COMPUTER SCIENCE COURSE ASSESSMENT WITH RESPECT TO ACM CORE CURRICULUM TOPICS

Authors

Matthew Gill - matthew.gill@selu.edu
Rachel Pierson - rachel.pierson@selu.edu

Faculty Sponsors

Kuo-Pao Yang - kyang@selu.edu
Theresa Beaubouef - tbeaubouef@selu.edu

Southeastern Louisiana University

College of Science and Technology
Department of Computer Science and Industrial Technology
SLU 10847
Hammond, LA 70402

ABSTRACT

This project is intended to allow instructors to create and manage college course topics in relation to the ACM core curriculum. Additionally, this project will generate an assessment form specific to each course, which will be filled out by the students enrolled in that course. The results will be compiled into a separate summary assessment that will enable the course instructor to improve his/her quality of teaching.

INTRODUCTION

Assessment is very important when faculty and education administrators consider adding or removing courses to a given curriculum. The nature in-which administrators acquire information pertaining to a given curriculum varies by institution, but inevitable must allow those participating and volunteering information to have an easy to use interface that feels natural to use while still communicating the information needed. This project makes use of various manners of interface design and coding practices that will allow instructor users to work effectively, and grant student users the facility of a quick, straight-to-the-point assessment [1,2,4,5,6,7,8,9,10,11,12].

In 1998 the Computer Society of the Institute for Electrical and Electronic Engineers and the Association for Computing Machinery created a joint task force concerned with updating existing undergraduate programs in computing. Through the use of knowledge focus groups the joint task force revised *Computing Curricula 1991* to incorporate a decade of past development. The numerous revisions were compiled into *Computing Curricula 2001* [3].

The ACM relies on knowledge focus groups (KFGs) to categorize similar individual course attributes. From an analytical standpoint, a computer science course could fall into a number of these groups. While each KFG has a minimum number of hours that must be met for the curriculum to qualify for that particular KFG, individual courses may incrementally acquire KFG hour points based on the material and length of the course. The 14 knowledge focus groups are:

Discrete Structures (DS)
Programming Fundamentals (PF)

Algorithms and Complexity (AL)
Architecture and Organization (AR)
Operating Systems (OS)
Net-Centric Computing (NC)
Programming Languages (PL)
Human-Computer Interaction (HC)
Graphics and Visual Computing (GV)
Intelligent Systems (IS)
Information Management (IM)
Social and Professional Issues (SP)
Software Engineering (SE)
Computational Science and Numerical Methods (CN)

Currently (as of Fall 2008) the Computer Science course assessment does not take into account the CC2001's KFGs, though most courses are compatible [3].

SOFTWARE PACKAGE

PHP is a general-purpose scripting language used to create dynamic web pages. PHP can also be embedded into HTML. This makes PHP an incredibly versatile and lightweight language. PHP's ability to connect and manipulate data over many webpages and within databases has allowed the development team to construct generic webpages filled with dynamic PHP code. Simply put, the same webpage is used for every page of its type. Figure 1.1 illustrates a PHP function that dynamically populates an HTML drop-down box with database information concerning courses.

```
$strNameField2 = "classNum";
$strNameField3 = "classSec";
$strTableName = "classes";
$iInstructor = $iUsername;
$strOrderField = "classNum";
$strMethod="asc";

echo "<select name=\"eClass\">\n";
echo "<option value=\"NULL\">Select Value</option>\n";

$strQuery = "select $intIdField, $strNameField1, $strNameField2, $strNameField3 from
    $strTableName WHERE iUsername='$iInstructor' order by $strOrderField $strMethod";
$rsrcResult = mysql_query($strQuery);

$strCompare = "";
while($arrayRow = mysql_fetch_assoc($rsrcResult)) {

    $strA = $arrayRow["$intIdField"];
    $strB = $arrayRow["$strNameField1"];
    $strC = $arrayRow["$strNameField2"];
    $strD = $arrayRow["$strNameField3"];
    $strComp = $strB . " " . $strC . " - " . $strD;
    if($strComp == $strCompare)
    {
        }
    else
    {
        echo "<option value=\"$strComp\">$strComp</option>\n";
        $strCompare = $strComp;
    }
}
echo "</select>\n\n";

?>
```

Figure 1.1
PHP Code Snippet

MySQL is a relational database management system that allows multiple users to edit information. SQL is a trusted standard in the computing industry concerning database management. Figure 1.2 illustrates the ‘classes’ table of a MySQL database.

	Field	Type	Charset	Attributes	Null	Default	Extra	Action					
<input type="checkbox"/>	id	int(4)			No		auto_increment						
<input type="checkbox"/>	wNum	varchar(8)			No								
<input type="checkbox"/>	classType	varchar(4)			No								
<input type="checkbox"/>	classNum	char(3)			No								
<input type="checkbox"/>	classSec	char(2)			No								
<input type="checkbox"/>	tSeason	varchar(10)			No								
<input type="checkbox"/>	tYear	varchar(4)			No								
<input type="checkbox"/>	taken	char(3)			No								

Figure 1.2
MySQL Table Structure

APPROACH

This project intends to replace the existing course assessment. The new assessment suite features a combination of PHP and MySQL. The development team has created a dynamic website using PHP that allows instructors to create, edit, remove, and analyze courses they teach. During the creation of a class, professors are given options to add assessment questions, and KFG requirements for a given course. When an instructor edits a course, they are able to change the KFG requirements and questions. Class analysis summarizes student answers to the assessment questions and creates a separate summary for the instructors. A student-only site is linked to the same MySQL database and allows students to choose their course and answer assessment questions. Extraneous programs or resources were available via Student Technology Fee computers.

The development team closely collaborated with the faculty sponsor and co-sponsor. Weekly meetings were organized to present the project’s progress and receive input from the faculty sponsors.

This project has allowed the development team an excellent opportunity to gain further experience with PHP and MySQL. The development team has built and enhanced skills that can be used professionally in the computing industry.

ROADBLOCKS

Roadblocks describe the issues encountered during the development of this project. PHP proved to be less flexible than languages in which project developers were proficient, thus resulting in a steep learning curve. Additionally, no extraneous software was used for the majority of this project. Microsoft Windows XP’s ‘Notepad’ was the primary editor used.

RESULTS

The completed project features administrative, instructor, and student modes. Modes, containing string variables, are sent to and from the web pages. Depending on the mode received, a single page is capable of displaying only what is required. For example, when logging into the student portion of the site, information regarding class and student information is sent to the main page along with the ‘student’ mode. The main page, upon receiving the mode ‘student’, displays either the desired assessment or an error message if the student has already taken the

assessment. The code for all available modes is present on the page, however, only the code for the received mode will be interpreted.

Administrative mode allows a special designation of user to manage instructor users. Instructor users are allowed to create, edit, remove, and analyze classes, as well as add students to classes. Additionally, all instructor users may view a numerical summarization of all Computer Science courses with respect to the Association for Computing Machinery's knowledge focus groups. Instructor users are able to define the learning objectives, minimum topics, assessment password, and ACM KFGs when creating and editing a class.

The screenshot shows a web interface for an instructor. At the top left, it says "login, test" and at the top right, "Log-Out". The main content area is titled "Create a Class:" and includes input fields for "Class Type:", "Class #:", "Section:", "Semester of Class:" (with a dropdown menu set to "SPRING"), and "Year of Class:". Below these is a "Create Class" button. The next section is "Edit a Class:" with a "Choose an existing class:" dropdown and an "Edit Class" button. This is followed by "Remove a Class:" with a similar dropdown and a "Remove Class" button. Then "Analyze a Class:" with a dropdown and an "Analyze Class" button. The "Add Students:" section has a dropdown for "Choose a class to add Students to:" and an "Add Students" button. At the bottom, there is a "Summarize Curriculum" button.

Figure 2.1
Instructor User Interface

Students are required to provide numerous pieces of information to certify they are in fact enrolled in a given course. Students must provide an ID number, which must be included in a list of IDs added by the course instructor, as well as a unique assessment password that the course instructor must provide. The student interface is simplistic and easy to follow, dynamic tables are generated from the Learning Objectives and Minimum Topics provided by the instructor. Students must then fill in radio buttons in response to a given question. See Figure 2.2.

The screenshot shows a student assessment form titled "Course Assessment" with a sub-header "Edited for Blind Review". It is divided into two sections: "Minimum Topics" and "Learning Objectives".

Minimum Topics

Question #	Question	Agree	NA	Disagree
Question 1:	This project was built using PHP?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Question 2:	This project was built using Perl?	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Learning Objectives

Question #	Question	Agree	NA	Disagree
Question 1:	Were you taught Java?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Question 2:	Were you taught PHP?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

At the bottom of the form is a "Submit Answers" button.

Figure 2.2
Sample Student Assessment

CONCLUSION & FUTURE WORK

The current status of the Computer Science course assessment is in dire need of revamping. It is out of date and does not contain any information relating the courses or curriculum to the Association for Computing Machinery's Knowledge Focus Groups. Using a combination of PHP and MySQL, this project has redesigned the entirety of how course assessment is administered. Instructors have the freedom to create, edit, remove, and analyze courses via administrative web pages. Additionally, students are able to answer assessment questions online, instantly giving feedback to the instructors.

Given more time, current security algorithms, user management systems, and the general graphical user interface would be enhanced. While the current GUI is sufficient, future work would have the GUI reworked to include custom logos and color schemes.

The security protocols implemented in the current version of the project are rather simple, thus enabling even a novice hacker to surpass the protocols altogether. Future work would constitute a complete reworking of instructor and administrative login methods as well as the implementation of password hashing. Currently, users are created using a unique administrative account. Future work would allow certain privileged instructors to create, modify, and remove users.

ACKNOWLEDGEMENTS

This project was supported by a Science and Technology Awards for Research (STAR) Grant from the College of Science & Technology, Southeastern Louisiana University.

REFERENCES

1. Barnes, H. and M. Gwyer. *Designer/2000 Web enabling your applications*, Oracle white paper. Technology Representative. Oracle Corp., Redwood City, CA, 1996.
2. Carroll, John M. *Human-computer interaction in the new millennium*, New York, NY: ACM Press; Boston, MA: Addison-Wesley, 2002.
3. Computing Curricula 2001, Computer Science. *Association for Computing Machinery*, <http://www.acm.org/education/education/curric_vols/cc2001.pdf>.
4. Gall, M.D., J.P. Gall, and W.R. Borg. *Educational Research, 7th Edition*. p. 198. Boston, MA: Pearson Education, Inc., 2003.
5. Kessler, M. *A schema-based approach to HTML authoring*. In *Proceedings of the Fourth International Conference on The World Wide Web* (Boston, MA), 1995.
6. Marghitu D., L. Jackson, and K. Chang. *Web-Based Software Integration of Skill Assessment and Testing IT Tools for the University Computer Competency Exams*. ACMSE 2003 Proceedings, Savannah, GA, 2003.
7. Nielsen, J. *Computer Science and Engineering Handbook*. CRC Press, Inc., Boca Raton, FL, 1996.
8. Paolo Atzeni, Giansalvatore Mecca, Paolo Merialdo, *To Weave the Web, Proceedings of the 23rd International Conference on Very Large Data Bases*, p. 206-215, August 25-29, 1997.
9. Paolo Atzeni, Giansalvatore Mecca, Paolo Merialdo, *Design and Maintenance of Data-Intensive Web Sites, Proceedings of the 6th International Conference on Extending Database Technology: Advances in Database Technology*, p. 436-450, March 23-27, 1998.

10. Schwabe, Daniel, Gustavo Rossi, *The object-oriented hypermedia design model*, *Communications of the ACM*, v.38 n.8, p. 45-46, Aug. 1995.
11. Siegel, D. *Creating Killer Web Sites*, 2nd Edition. Indianapolis, IN: Hayden Books, 1997.
12. Stamey, J.W. and B.T. Saunders. *Unit Testing and Debugging with Aspects*. In Proceedings of ACM CCSC-NE, 2005.