

# A Web-Based System for Course and Program Assessment: Prototype to Product

Sofya Poger  
Computer Science Department  
Iona College  
New Rochelle, NY 10801, USA  
[spoger@iona.edu](mailto:spoger@iona.edu)

Frances Bailie  
Computer Science Department  
Iona College  
New Rochelle, NY 10801, USA  
[fbailie@iona.edu](mailto:fbailie@iona.edu)

**Abstract:** This paper describes how a prototype web-based system developed in the Computer Science Department of Iona College for on-line assessment of course objectives and prerequisites has been implemented as a product for essential assessment and accreditation purposes. The system was piloted by faculty and students for a year, gathering proposed modifications. As part of the software cycle, these suggestions were implemented to release a usable system to fulfill departmental, institutional and accrediting requirements. The system is currently used to assess our ABET accredited BS in Computer Science program. Plans are to extend coverage to assess our NCATE accredited MS in Educational Technology program. This automated system is designed to simplify faculty assessment responsibilities while meeting accreditation standards.

## The Prototype System

Course and program assessment has become a priority for institutions of higher education as they attempt to meet the criteria of their regional accrediting bodies and accrediting organizations of specific disciplines. Stakeholders such as boards of trustees, students, parents, employers and graduate schools must be assured that an institution and its individual departments are delivering what they promise in terms of student outcomes. Assessment can be a tedious process in which faculty are often reluctant to participate on a regular basis. Establishing a useful assessment process involves two key components: first, to develop a user-friendly system that will actually be used and second, to create results that can be easily and accurately interpreted for program improvement. In today's technological environment, it seems appropriate that assessment materials be gathered electronically through a web-based system.

As a result of a recent Middle States visit, Iona College has mandated that departments develop a systematic method for course and program assessment. The Computer Science Department at Iona College offers two programs that can be accredited: an MS in Educational Technology accredited by The National Council for Accreditation of Teacher Education (NCATE) and a BS in Computer Science accredited by The Accreditation Board for Engineering and Technology (ABET). Both of these accrediting agencies require assessment procedures that give evidence of student learning and validate program and course objectives. In 2003, the Department chose to target the BS program in Computer Science for the prototype assessment system. The first step was to develop a Department Mission statement drawn from the College Mission Statement and the Mission Statement of the School of Arts and Science. Next, the department laid out a set of program objectives for the BS program in Computer Science. Working from the program objectives, the Department fleshed out prerequisite skills and course objectives for each course in the program and mapped course objectives to program objectives (Poger, Kamari, et al , 2005).

The Assessment Committee then established a set of assessment tools and subsequently designed a prototype system to collect data and report results to target problem areas and plan for course and program improvement. The system would be required to compile data from the following sources:

1. **Prerequisite Skills Student Survey.** By the fourth week of each semester, students would complete a survey to determine how well they had mastered the specific prerequisite skills of each course. These prerequisite skills were drawn from the course objectives of the prerequisite courses.
2. **Course Objectives Student Survey.** At the end of each semester, students would complete a survey to determine how well they had mastered the specific course objectives of each course.
3. **Faculty Course Objectives Assessment.** For each assessment administered in a course (test, project, paper, and so on), the faculty member would determine which of the course objectives the assessment was designed to meet. Then the faculty member would evaluate how well the students had met each objective of the assessment. By the end of the course, it would be possible to determine how well the faculty member believed the students had mastered each of the course objectives
4. **Graduating Student Exit Survey:** As students completed the program, they would complete a survey to determine how well they had mastered the program objectives.

Graduate and undergraduate students enrolled in a capstone project course were assigned to implement the system. Student groups were formed to work on the individual modules of the system. Working with the first author and the system administrator, modules were released for use as they became available. Reports were generated from the prerequisite skill survey, the course objectives student survey and the faculty course objectives assessment. The Assessment Committee used these reports to identify problems that were brought to the full department for discussion and resolution.

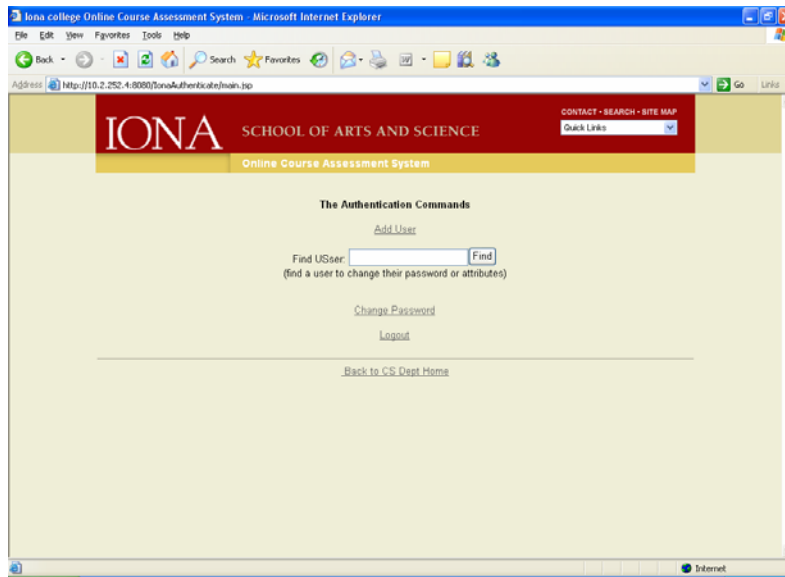
## **System Modifications**

The alpha version of this system was installed in September 2004. Reports from the system were vital in compiling assessment data for the Department's ABET Accreditation visit in Fall 2004. Faculty and students used the system during the 2004-2005 academic year. Assessment is an on-going process, and the Department was well-aware that reports generated for the accreditation visit would have to be continued. It soon became clear that the alpha version of the system required enhancements to support on going faculty use.

“Software development projects are subject to constant change. . . . they need to be updated when errors are discovered and when the developers have a better understanding of the application.” (Brwegge & Dutoit, 2004, p. 5) A key element in the software development cycle is feedback from the users that can be used to improve not only the system interface but also the integrity and usefulness of the output data. Suggestions were collected from all constituents and a new set of students was recruited to put into place the following vital modifications:

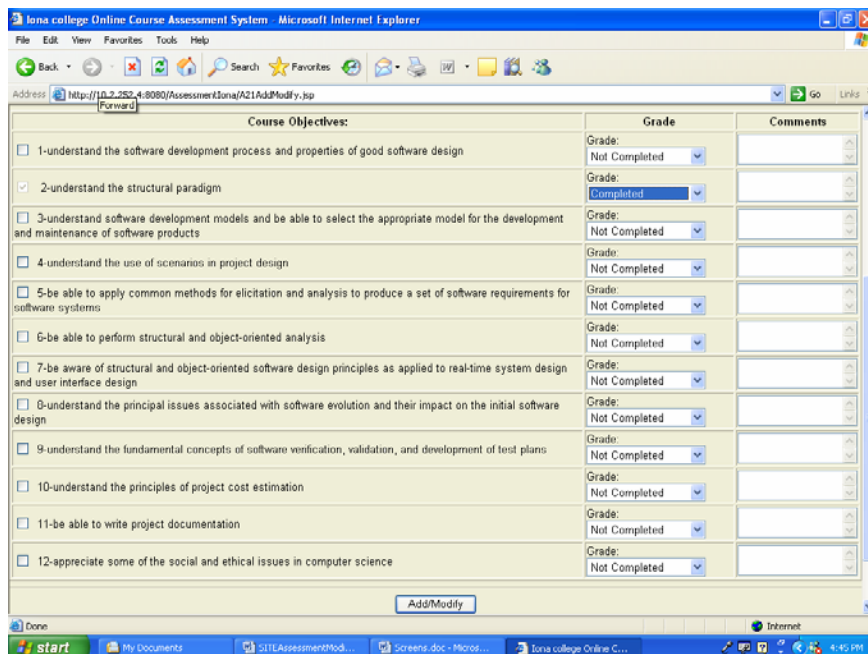
1. **The Security System.** It is essential that a web-based system secure user input and maintain proper access privileges (e.g., administrators, faculty, students, and so on). In the alpha system, the passwords were stored in a database as plain text and thus were vulnerable to misuse. The solution employed in the beta system was to encrypt the passwords with a hashing algorithm and to keep the user name and password pair in an application isolated from the assessment system. This application would be responsible for safeguarding the password and allowing administrators privileges for system maintenance functions such as adding users to the system and changing passwords. Figure 1 illustrates some of the functions in the new security system.

Figure 1



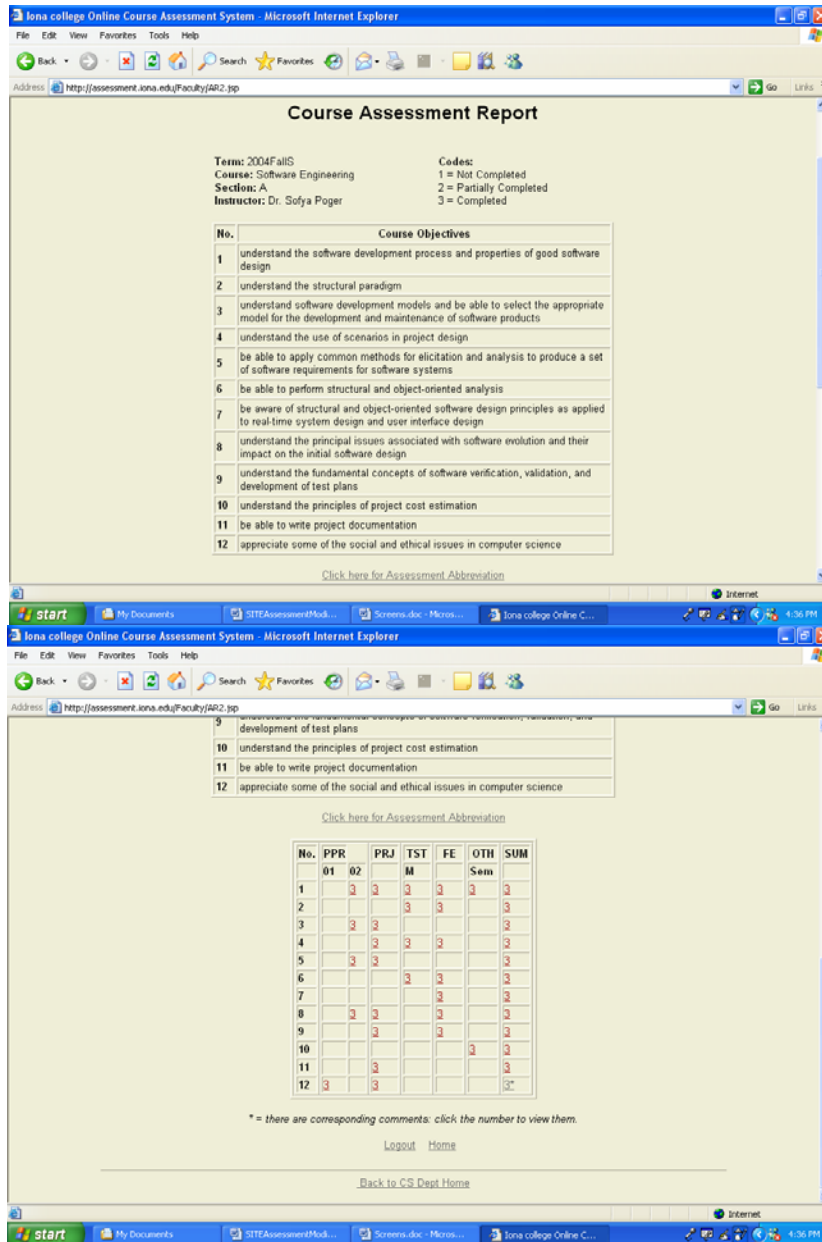
- Faculty Course Objective Assessment:** The alpha system required selecting a course objective and then rating all the assessments associated with the objective. In practice, this method was awkward since faculty were interested in rating how well each assessment met its objectives. It is often difficult to convince faculty to perform assessment procedures, so a system that was counter-intuitive was not likely to be utilized. In response to this problem, the beta system allowed the faculty to select an assessment and then rate all the objectives the assessment targeted. Another enhancement was the ability to display the ratings for all the objectives, thus identifying any unused or unmet objectives. Figure 2 illustrates these improvements.

Figure 2



3. **Report Format:** In an effort to produce output from the faculty assessments in a timely fashion, little attention was given to the organization of the output in the alpha system. Reports spanned multiple pages and were difficult to interpret. The beta system redesigned the output in a meaningful format so that important summary information could be easily captured. Figure 3 depicts the modified report.

Figure 3



4. **Program Objectives:** In preparing accreditation materials, it was necessary to identify specific course assessments that would be used to assess program objectives. For example, course final project grades might be a measure of how well students had mastered a specific program objective. This requirement had not been part of the alpha system. The beta version now maps these program objectives to their specific courses so they can be properly evaluated by students and faculty.

5. **Input data:** For the system to function, tables must be populated with specific course information and class rosters for each course. In the alpha system, students inputted the course information and faculty entered the students registered in each class. To save time and improve data integrity, the beta system is designed to interface with Iona's PeopleSoft System to obtain accurate course and student information.
6. **Graduating Student Exit Survey:** The alpha version did not implement the exit surveys, so they were administered and analyzed by hand. The beta version now allows graduating students to complete this survey online.
7. **Statistical Report Format:** Each of the student surveys and the faculty assessment produced statistical reports in the alpha version that were not user-friendly. Improved readability of these reports is an important feature of the beta version, as can be seen in Figure 4.

Figure 4

Online Course Evaluation System Statistics

Wed Oct 05 15:53:37 EDT 2005

FAC	COURSE	TERM	COURSE OBJECTIVE	#	AVG	MED	STD
Poger	CS481 A	2004FallS	177: understand the software development process and properties of good software design	10	4.40	5	1.07
Poger	CS481 A	2004FallS	170: understand the structural paradigm	10	3.50	4	1.10
Poger	CS481 A	2004FallS	179: understand software development models and be able to select the appropriate model for the development and maintenance of software products	10	4.30	4.5	.02
Poger	CS481 A	2004FallS	180: understand the use of scenarios in project design	10	4.50	5	.05
Poger	CS481 A	2004FallS	181: be able to apply common methods for elicitation and analysis to produce a set of software requirements for software systems	10	4.20	4.5	1.03
Poger	CS481 A	2004FallS	182: be able to perform structural and object-oriented analysis	10	4.30	4.5	.82
Poger	CS481 A	2004FallS	183: be aware of structural and object-oriented software design principles as applied to real-time system design and user interface design	10	4.30	4	.67
Poger	CS481 A	2004FallS	184: understand the principal issues associated with software evolution and their impact on the initial software design	10	4.10	4	.88
Poger	CS481 A	2004FallS	185: understand the fundamental concepts of software verification, validation, and development of test plans	10	4.40	4	.52
Poger	CS481 A	2004FallS	186: understand the principles of project cost estimation	10	3.00	4	1.14
Poger	CS481 A	2004FallS	187: be able to write project documentation	10	4.30	5	1.06
Poger	CS481 EA	2004FallS	180: appreciate some of the social and ethical issues in computer science	10	4.40	5	.97
Poger	CS481 EA	2005SpringS	177: understand the software development process and properties of good software design	2	5.00	5	.00
Poger	CS481 EA	2005SpringS	178: understand the structural paradigm	2	4.50	4.5	.71
Poger	CS481 EA	2005SpringS	179: understand software development models and be able to select the appropriate model for the development and maintenance of software products	2	4.50	4.5	.71
Poger	CS481 EA	2005SpringS	180: understand the use of scenarios in project design	2	4.50	4.5	.71

## Future Plans

Future plans for the Assessment System are for continued review and extension to other programs. It is essential to the life of any useful software system to continually evaluate the functionality and flexibility of the system by incorporating suggestions from system users and administrators. It is clear that the modifications implemented in this beta version will not remain static but grow in response to future requirements. The Department was awarded accreditation from ABET and, since assessment is a pivotal component of the accreditation process, the Department is eager to adapt the system to its NCATE accredited MS in Educational Technology program. From there it is hoped that the system can be extended first to the Department's other undergraduate and graduate programs. Following that, the system design is flexible enough to meet the assessment needs of other disciplines. Assessment procedures are an absolute necessity in higher education and the demand for a user-friendly automated system that produces reliable and useful data will be present for the foreseeable future.

## Literature References

Abunawass, A., Lloyd, W., & Rudolph, E. (2004). On-line Feedback and Assessment: COMPASS: A CS Program Assessment System. Proceedings of the 9<sup>th</sup> Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, *ACM SIGCSE Bulletin* 36 (3), 127-131.

- Accreditation Board for Engineering and Technology, ABET Computing Accreditation Commission (2003). *Criteria For Accrediting Computing Programs*. ABET
- Afgouin, H. (2006) *Database Security and Auditing*. Thompson Course Technology
- Blaha, K. & Murphy, L. (2001). Targeting assessment: how to hit the bull's eye. *Journal of Computing Sciences in Colleges* 17(2), 112-121.
- Blanford, D. & Hwang, D. (2003). Five Easy but Effective Assessment Methods. Proceedings of the 34<sup>th</sup> SIGCSE Technical Symposium on Computer Science Education, *ACM SIGCSE Bulletin* 35(1), 41-44.
- Brennan, J. & Shaw, T. (2000). *Managing quality in higher education: an international perspective on institutional assessment and change*. Open University Press.
- Brwegge, B. & Dutoit, A.H. (2004). *Object-Oriented Software Engineering Using UML, Patterns and Java*, 2<sup>nd</sup> edition. Prentice Hall.
- Caristi, J., Dale, N., Marion, B. & Turner, J. (1994). Assessment in Computer Science. Proceedings of the 25<sup>th</sup> SIGCSE Technical Symposium on Computer Science Education, *ACM SIGCSE Bulletin* 26(1), 401-402.
- Daniels, M. Berglund, A. Pears, A. & Fincher, S. (2004). Five Myths of Assessment. Proceedings of the sixth conference on Australian Computing Education, (30), 57-61.
- Diamond, R. (1998). *Designing and Assessing Courses and Curricula: A Practical Guide*. Jossey-Bass.
- Falkner, J. et.al. (2001). *Beginning JSP Web Development*, Wrox Press.
- Falkner, J. and Jones, K. (2004). "Servlets and JavaServer Pages", Addison Wesley.
- Hernon, P. & Dugan, R. (Ed.) (2004). *Outcomes Assessment in Higher Education: Views and Perspectives*, Libraries Unlimited.
- Hogan, T., Harrison, P. & Schultze, K. (2002). Developing and Maintaining an Effective Assessment Program. *ACM SIGCSE Bulletin* 34(4), 52-56.
- McCracken, D.D. & Wolfe, R. (2004). *User-Centered Website Development: A Human-Computer Interaction Approach*. Prentice Hall.
- Palomba, C. & Banta, T. (2001). *Assessing Student Competence in Accredited Disciplines: Pioneering Approaches to Assessment in Higher Education*, Stylus Pub.
- Poger, S., Schiaffino, R., Ricardo, C. (2005) "A Software Development Project: A Student-written Assessment System", *The Journal of Computing Sciences in Colleges* 20(5), pp. 229-238.
- Poger, S., Kamari, A., Chuah, Y., Ricardo, C. (2005). "Development of a Web-Based System for Course and Program Assessment", *Association for the Advancement of Computing in Education (AACE) Proceeding of the Society for Information Technology and Teacher Education (SITE) Annual Conference, Phoenix, AZ* pp.176-182.
- Sanders, K. & McCartney, R. (2003). Program Assessment Tools in Computer Science: A Report from the Trenches. Proceedings of the 34<sup>th</sup> SIGCSE Technical Symposium on Computer Science Education, *ACM SIGCSE Bulletin* 35(1), 31-35.
- Steelman, A. and Murach, J. (2001). "Murach's Java Servlets and JSP", Mike Murach & Associates, Inc.
- Urman, S. (2002). *Oracle 9i PL/SQL Programming*. Oracle Press, McGraw-Hill Osborne.
- Whiteman, M.E. & Mattord, H.J (2005), *Principles of Information Security*, 2<sup>nd</sup> edition. Thompson Course Technology.