An Accreditation Framework for Global Markets

Robert Kerbs
California State Polytechnic University, Pomona, Computer Science Department,
Pomona, CA 91768 rwkerbs@csupomona.edu

Abstract - **The undergraduate computer science program at Cal Poly Pomona has been accredited since 1994. In our most recent ABET General Review, we developed a new assessment framework that takes into account industry needs in the Los Angeles area, student learning requirements, and programmatic needs. It was necessary to create this new framework in response to the changing nature of our local industry. We have experienced the dot-com bubble burst and a contraction in the aerospace industry, both of which have resulted in fewer entry-level jobs for our graduates. At the same time, we have witnessed an increase in the number of companies that operate globally in our area. To adapt to this changing environment, we use three tools in our assessment framework: course-based student portfolios, graduating student surveys, and input from advisory board meetings with industry partners. Results obtained with these tools allow us to revisit our educational objectives and modify our desired program outcomes, ultimately allowing us to make changes to our courses that reflect industry needs. This accreditation framework has insured continual program improvement. Our development and experience with this accreditation framework are described in this paper.**

Index Terms – Accreditation, Assessment, Curriculum revision, Outcomes-based learning.

INTRODUCTION

Cal Poly Pomona is one of two polytechnic universities in the 23-school California state university system. Our undergraduate computer science (CS) program has been continuously accredited since 1994. The 2002-2003 accreditation cycle was our most recent ABET General Review. At the time, the CS faculty was overwhelmed with undergraduates entering our major. At our peak, we had more than 1200 computer science majors covered by 13 tenure-track faculty members. This required that we employ a pool of part-time instructors to adequately help provide coverage of the curriculum. During the 2001-2002 academic year the undergraduate program was impacted to limit the number of incoming students. Consequently, ABET's visiting team identified some weaknesses associated to these events which required the design and implementation of a comprehensive assessment program to insure continual program improvement.

During this time-frame we began to see evidence of the dot-com bubble burst, a compaction of the aerospace industry in our area, and we began to read that certain programming tasks were being outsourced overseas for between $1/5^{th}$ to $1/10^{th}$ of the cost in the United States [1]. Our undergraduate CS program began to see a dramatic decrease in the number of incoming freshman and transfer students. Feedback from our industry advisory board informed us that companies were severely limiting the number of their fresh CS graduate hires but instead giving preference to those with approximately five years industry experience. In addition, some advisory board members said that they if they did have a position for a recent CS graduate, the candidate would only be considered if she or he secured and completed an internship via the company.

This paper discusses Cal Poly Pomona's CS department's assessment program and how it is being used to adapt to the changing industrial environment. Background information is presented followed by the CS department's assessment framework. The program's choice of assessment tools is introduced as well as a discussion of faculty involvement. Finally, a discussion and future work section closes the paper.

BACKGROUND

Cal Poly Pomona's computer science department currently has 10 full-time tenure-track, three adjunct, and three half-time faculty members who are enrolled in the university's Faculty Early Retirement Program (FERP). We serve approximately 500 undergraduate students and 50 graduate students.

Our undergraduate program offers a core sequence of classes followed by a wide-range of upper-division elective classes. We considered moving to track-based curriculum for the upper-division courses but found, for the moment, that students would have more flexibility picking and choosing their electives. Additionally, some faculty members think scheduling would have been too constrained resulting in classes not being offered with sufficient frequency for students to complete their degrees in four years.

The undergraduate CS program has been continuously accredited since 1994. We believe being accredited is valuable in keeping the quality of the program high. As others have found [2], accreditation is good for marketing our program to students, parents, and industry.

ACCREDITATION

At the end of 2002, we had an ABET General Review and site visit. The visiting team identified deficiencies that required immediate attention to assure reaccreditation. A common area that programs such as ours fall victim to is not keeping adequate documentation [3]. While we felt we made representative copies of student work, the visiting team could find no evidence that we were adequately covering social and ethical issues in computing. A more serious and foundational problem was our program didn't adequately "close the loop [4]," see figure 1.

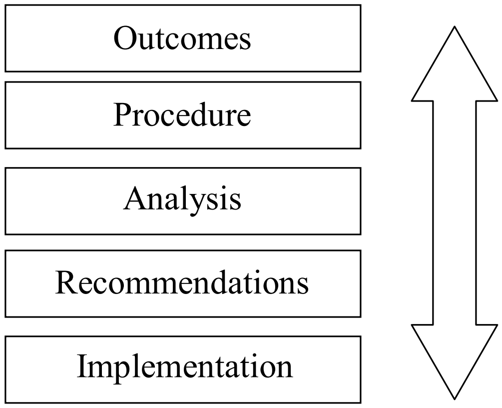

FIGURE 1
CLOSING THE LOOP

ABET accreditation is outcomes-based [5][6][7]. We should have had expected outcomes associated with our courses that articulated what the students could do after successfully completing a given course. Next, we needed to have a routine procedure in place to collect our assessment data. Up to this point in time, it appeared as if we collected our materials in an ad-hoc manner. Next, there needed to be a mechanism to periodically analyze the data. From the analysis, recommendations for program improvement could have been made. Finally, implementation of the recommendations would have closed the loop.

After we received ABET's final statement we constructed a response and have subsequently submitted two interim reports. Needless to say, our program was forced to re-examine itself and to establish a comprehensive assessment program to address these issues. A task force was assembled that included the CS chair, a former CS chair, and myself. We attended an ABET faculty workshop sponsored by the engineering department in order to gain further insight into the ABET's outcomes-based accreditation process. The administration supported our efforts by providing release-time support for an assessment coordinator.

ASSESSMENT FRAMEWORK

We began establishing our assessment framework by examining Cal Poly Pomona's Mission Statement: "Cal Poly Pomona's mission is to advance learning and knowledge by linking theory and practice in all disciplines, and to prepare students for lifelong learning, leadership, and careers in a changing, multicultural world." The university's motto is "learn by doing." Using this as a basis, we began establishing educational objectives for the students, table I.

TABLE I
EDUCATIONAL OBJECTIVES

EO1.	A solid foundation in the fundamental areas of Computer Science
EO2.	A good grasp of both hardware and software aspects of computer systems
EO3.	A balance between theoretical and practical aspects of computing
EO4.	Incorporation of campus "Learn by Doing" philosophy into all aspects of the program
EO5.	Awareness of discipline-related ethics and social responsibility issues
EO6.	Training as a software professional and preparation for lifelong learning

In order to support the educational objectives, we established a series of CS program outcomes, see table II. These program outcomes were influenced by ACM/IEEE curriculum guidelines.

TABLE II
CS PROGRAM OUTCOMES

PO1.	Proficiency in algorithms, data structures, software design, programming language concepts and computer architecture
PO2.	Proficiency in theoretical foundations of computing, problem analysis and solution design
PO3.	Proficiency in one or more modern programming languages
PO4.	Proficiency in oral and written communication
PO5.	Understanding of social and ethical issues in computing

Table III shows the relationship between the student educational objectives and the CS program outcomes.

TABLE III
RELATIONSHIP OF STUDENT EDUCATIONAL OUTCOMES TO CS PROGRAM OUTCOMES.

	PO1	PO2	PO3	PO4	PO5
EO1	X	X	X		
EO2	X	X	X		
EO3	X	X	X		
EO4	X	X	X	X	
EO5				X	X
EO6	X	X	X	X	X

As with other programs [8], we needed to insure that our educational objectives and CS program outcomes were supported by specific courses. We refer to these courses as "witness courses," see table IV.

The overall relationship among educational objectives, CS program outcomes, and witness courses is shown in Table V.

TABLE IV
WITNESS COURSES

CS 130	Discrete Structures
CS 140	Introduction to Computer Science
CS 141	Introduction to Programming and Problem Solving
CS 210	Computer Logic
CS 240	Data Structures and Algorithms I
CS 241	Data Structures and Algorithms II
CS 301	Numerical Methods
CS 310	Formal Languages
CS 331	Design and Analysis of Algorithms
CS 420	Artificial Intelligence
CS 431	Operating Systems
CS 435	Database Systems
CS 440	Compiler Design
CS 463	Undergraduate Seminar

TABLE V
OVERALL RELATIONSHIP

	EO1	EO2	EO3	EO4	EO5	EO6	PO1	PO2	PO3	PO4	PO5
CS130	x	x	x	x		x		x			
CS140					x	x					x
CS141	x	x	x	x		x			x		
CS210	x	x	x	x		x	x				
CS240	x	x	x	x		x	x				
CS241	x	x	x	x		x			x		
CS301	x	x	x	x		x	x				
CS310	x	x	x	x		x		x			
CS331	x	x	x	x		x	x	x			
CS420	x	x	x	x	x	x		x		x	
CS431				x	x	x				x	
CS435	x	x	x	x	x	x	x				x
CS440	x	x	x	x		x	x	x	x		
CS463				x	x	x				x	x

ASSESSMENT TOOLS

ABET provides sufficient flexibility for a program to tailor its particular assessment approach based upon its own unique constraints. We determined that we needed to obtain both internal and external feedback regarding our program. Of the many assessment tools available [8][9], we chose: course-based student portfolios, graduating student surveys, and input from our industry advisory board.

Course-based student portfolios

This assessment tool is a somewhat underutilized resource due the labor involved in collecting materials and the time commitment required to analyze them [9]. As with other institutions [10], we have found them to directly correlate programmatic outcomes to our curriculum. Consequently, we have found them to be a good fit for our program.

Each quarter, instructors in selected witness courses build student portfolios based on a learning outcomes template customized for the course, see figure 2. The course outcomes template is developed with input from, and the consensus of, instructors currently teaching the course. At the end of the quarter, the instructor uses the portfolios and the course outcomes list template to measure student learning in that course relative to the various program outcomes it witnesses, see figure 3. Additionally, each instructor prepares a portfolio summary report detailing findings and suggestions for improvement. Once a year, all instructors who have taught a given witness course meet to examine and discuss report findings and portfolios. Current outcomes are discussed and if any changes need to be made are brought to the curriculum committee's attention.

Graduating student surveys

These surveys are conducted in the capstone courses CS 440 (Compiler Design) and CS 463 (Undergraduate Seminar) each quarter. Both quantitative responses relating to the program objectives and qualitative responses seeking input about the pros, cons and suggestions for improvement are solicited. Once per year all surveys are consolidated and analyzed.

Industry advisory board

Annual industry advisory board meetings are arranged with representatives of firms that hire our students and alumni professionals who can offer feedback and direction to our program. Advisory board minutes summarizing comments and suggested improvements are prepared and posted on the department's accreditation website

Results obtained with these tools allow us to revisit our educational objectives and modify our desired CS program outcomes, ultimately allowing us to make changes to our courses that reflect industry needs.

FACULTY INVOLVEMENT

Faculty recognize that assessment tool utilization and analysis of results is a time-consuming activity and ways to minimize the time-commitment is appreciated [10][11]. With that said, faculty are more open to contribute to an educational program if they are able to articulate what it is they think students should learn in their courses [5]. To help with this issue, we followed the lead of other programs [12] and, with the support from the administration, established assessment coordinator position to serve as focal point for assessment-related activities in the department.

DISCUSSION AND FUTURE WORK

The Bureau of Labor Statistics expects *computer scientist* to be among the fastest growing occupations through 2014 [13]. However, due to the nature of outsourcing, the dot-com bubble-burst, and the subsequent decline in the aerospace industry in our area, it is especially important that we continue to communicate with industry in order to evolve the CS program in a way that supports the university's mission. While we understand that outsourcing of critical components is hard to do, we do recognize that as workers in other countries gain expertise and experience, these components might too be

Cal Poly Pomona Computer Science Department

Course Assessment Form for CS 140/141

CATALOG DESCRIPTION(S)

CS 140 Introduction to Computer Science Basic concepts of Computer Science, including hardware and software. Ethical and social impacts of computing. Problem-solving methods. Programming in an object-oriented language. Written essay required. 4 lectures/problem-solving. Prerequisite: Eligibility for MAT 114 or consent of instructor.

CS 141 Introduction to Programming and Problem-Solving Design, implementation, documentation and testing of programs in an object-oriented language. Modularization and reusability of software. File I/O, graphic user interfaces, and exception handling. 4 lectures/problem-solving. Prerequisite: CS 140 and MAT 114 with a grade of C or better, or consent of instructor. credit limited to 8 units, with a maximum of 4 units per quarter. Instruction is by lecture, laboratory or a combination. Prerequisite: consent of instructor.

Please indicate which of the following learning outcomes you expect *most students* in your class to meet. In the EVIDENCE column, list the *specific* instruments (e.g., Quiz 3, Homework 5, etc.) that you will use to provide evidence that the outcome has been met:

	EXPECT	DO NOT EXPECT	EVIDENCE	PO-LIST
1. Familiarity with ethical and social issues involving computers.	❑	❑	________	4,5
2. Understanding of the history of computing and the landscape of programming languages.	❑	❑	________	1,2
3. Competence at using a command-line interface.	❑	❑	________	1
4. Competence at using basic elements of the Java language including primitive data types, operators, and control structures for selection and iteration.	❑	❑	________	1,3
5. Ability to use predefined classes and methods.	❑	❑	________	1,2,3
6. Ability to define new classes and instances thereof.	❑	❑	________	1,2,3
7. Ability to read and write to files and to the console.	❑	❑	________	2,3
8. Competence at testing computer programs, including the use of exceptions.	❑	❑	________	2,3
9. Competence at basic algorithms involving collections.	❑	❑	________	1,2,3
10. Understanding of simple sorting and searching.	❑	❑	________	1,2,3
11. Competence at inheritance, class hierarchies and polymorphism.	❑	❑	________	1,2,3
12. Understanding simple recursion.	❑	❑	________	1,2,3

FIGURE 2
SAMPLE LEARNINGS OUTCOME TEMPLATE

Please examine each student portfolio to assess the extent to which each of the following student learning outcomes appears to have been met by this student:

	MET				NOT MET
1. Familiarity with ethical and social issues involving computers.	5	4	3	2	1
2. Understanding of the history of computing and the landscape of programming languages.	5	4	3	2	1
3. Competence at using a command-line interface.	5	4	3	2	1
4. Competence at using basic elements of the Java language including primitive data types, operators, and control structures for selection and iteration.	5	4	3	2	1
5. Ability to use predefined classes and methods.	5	4	3	2	1
6. Ability to define new classes and instances thereof.	5	4	3	2	1
7. Ability to read and write to files and to the console.	5	4	3	2	1
8. Competence at testing computer programs, including the use of exceptions.	5	4	3	2	1
9. Competence at basic algorithms involving collections.	5	4	3	2	1
10. Understanding of simple sorting and searching.	5	4	3	2	1
11. Competence at inheritance, class hierarchies and polymorphism.	5	4	3	2	1
12. Understanding simple recursion.	5	4	3	2	1

FIGURE 3
TEMPLATE MEASURING THE DEGREE TO WHICH A STUDENT MET AN EACH OUTCOME FOR A COURSE

outsourced [1]. The good news for our program is that we now have a mechanism to monitor and periodically update the program. The challenge we have is that we need to constantly refresh our advisory board based upon what jobs are available. This highlights the need for academia and industry to forge strong ties when possible.

ACKNOWLEDGMENT

Thanks are due to Dr. Mandayam Srinivas, Associate Dean, and Dr. Barry Soroka, Professor, College of Science at Cal Poly Pomona. We worked as a team to create and implement this assessment framework.

REFERENCES

[1] Hoganson, K, "Computer Science Curricula in a Global Competitive Environment," *Journal of Computing Sciences in Colleges,* 20, 1., October 2004, 168-177.
[2] Bailie, F, K., Abunawass, A, M., Houle, B, Whitfield, D, "Guidelines and Suggestions for ABET Accreditation," *Journal of Computing Sciences in Colleges,* 21, 2., December 2005, 83-85.
[3] Carey, E, L., "A Quest for ABET Accreditation: In Retrospect," *Journal of Computing Sciences in Colleges,* 19, 1., October 2003, 139-146.
[4] Maxim, B, R., "Closing the Loop: Assessment and Accreditation," *Journal of Computing Sciences in Colleges,* 20, 1., October 2004, 7-18.
[5] Crouch, D, B., Schwartzman, L, "Computer Science Accreditation: The Advantage of Being Different," *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education,* 2003, 36-40.
[6] Yaverbaum, G, Lidtke, D, Reichgelt, H, Zweben, S, "Outcomes-based Computing Accreditation: Program Assessment," *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education,* 2007, 281-282.
[7] Rigby, S, Dark, M, "Using Outcomes-based Assessment Data to Improve Assessment and Instruction: A Case Study," *ACM SIGITE Newsletter,* 3, 1., 2006, 10-15.
[8] Rao, M, R, K, K., Junaidu, S, Maghrabi, T, Shafique, M, Ahmed, M, et al., "Principles of Curriculum Design and Revision: A Case Study in Implementing Computing Curricula CC2001," *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education,* 2005, 256-260.
[9] Sanders, K, E., McCartney, R, "Program Assessment Tools in Computer Science: A Report from the Trenches," *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education,* 2003, 31-35.
[10] Blandford, D, K., Hwang, D, J., "Five Easy but Effective Assessment Methods," *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education,* 2003, 41-44.
[11] Yue, K, "Effective Course-based Learning Outcome Assessment for ABET Accreditation of Computing Programs," *Journal of Computing Sciences in Colleges,* 22, 4., April 2007, 252-259.
[12] Hogan, T, R., Harrison, P, R., Schulze, K, G., "Developing and Maintaining an Effective Assessment Program," *ACM SIGCSE Bulletin,* 34, 4., December 2002, 52-55.
[13] Bureau of Labor Statistics, "Occupational Outlook Handbook (OOH), 2006-2007 Edition," http://www.bls.gov/oco, accessed March 1, 2007.