

# Continuous improvement of the Computer Science Undergraduate Program at CSULA

Russ Abbott, Vladimir Akis, Valentino Crespi, Huiping Guo\*,  
Jiang Guo, Eun-Young Kang, Raj Pamula, Behzad Parviz, Chengyu Sun  
Department of Computer Science  
California State University, Los Angeles  
(hpguo@calstatela.edu)

## Abstract

*The Department of Computer Science at California State University, Los Angeles has established an assessment process to guide the continuous improvement of the undergraduate curriculum. The assessment results are used to identify program modifications. This paper discusses the results of such a process over a five-year period.*

## 1. Introduction

The Computer Science Department at California State University, Los Angeles implemented an assessment plan in 2001[1]. The assessment process involved significant steps that the faculty had to undertake:

- Identify Constituents (University Mission, Faculty, Students, Alumni and various accreditation agencies [2,3])
- Establish Vision (identifies where we want to be) and Mission (description of what we do) statements that conform to the statements made by the university.
- Establish Objectives, Student Learning Outcomes and Measures.
- Collect and analyze assessment data from various Measures.
- Feedback improvements to the curriculum.

The underlying philosophy has been to determine the extent to which the desired course goals have been met and how it leads to the successful attainment of a university's mission. The links connecting the university mission to our Course Goals can be best portrayed as a pyramid structure as shown in Figure 1.

- The broadest item, the Mission of the university, is at the top of the pyramid.
- The next two levels are the Mission Statements of the college and the department, which are derived from the university Mission Statement.
- The next level consists of the department Objectives, which when achieved, result in the accomplishment of the department Mission. Of our ten specified Objectives, two are direct educational Objectives.
- The next level consists of the measurable Student Learning Outcomes. When the Student Learning Outcomes are achieved, the educational objectives of the department are accomplished. We have specified six Student Learning Outcomes.
- The most detailed level consists of the Course Goals. Each Course Goal contributes to the satisfaction of one or more of the Student Learning Outcomes. Thus, meeting the goals of the courses in the program results in achieving the Learning Outcomes.

The assessment process can be defined as a two-loop approach as indicated in Figure 2. Loop #1 shows how the Learning Outcomes that support the program's educational Objectives are developed and evaluated. Loop#2 shows the steps involved in establishing Objectives and Learning Outcomes that are developed from various constituents. In either of the two loops, any identified course/program modifications are proposed and implemented.

Loop #1 activities were carried out once in the year 2001 and will be repeated at least once each five years. The department will monitor constituents' input and feedback to determine when this activity must be undertaken. Loop #2 activities are carried out

---

\* Corresponding author.

each year and ensure the continuous improvements of the program.

The assessment process has been well documented in [1] leading to the Self Study Report



Figure 1

## 2. Assessment Results

Assessment data is summarized below for each Learning Outcome.

- **Learning Outcome #1:**

Students will have a broad understanding of computing at all levels of abstraction.

- Graduating students will have a strong foundation in the design, analysis, and application of many types of algorithms.
- Graduating students will have a fundamental understanding of computer architecture and Operating Systems.
- Graduating students will have a fundamental understanding of automata theory.
- Graduating students will have a fundamental understanding of programming language paradigms
- Graduating students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.
- Graduating students will have fluency in at least on operating system and acquaintance with at least two more.

Analysis:

- The correlation between the MFT and student GPAs is greater than 80%. In other words our students seem to do as well in the MFT as they have done in the major (See Figure 3.)

[4] in preparing of ABET [2] accreditation. All our annual assessment reports are posted in [4]. In this paper, we discuss the assessment results [Section 2] leading to the program modifications [Section 3].

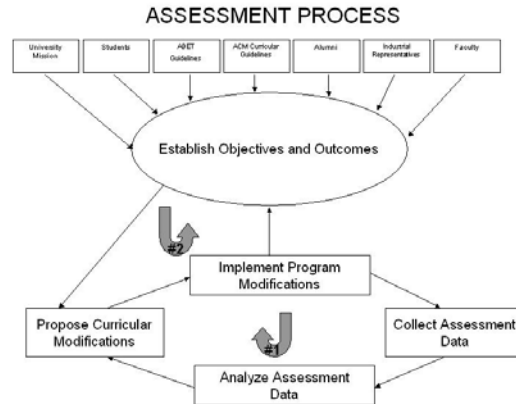


Figure 2

- Compared with students from over 200 colleges and universities taking the MFT, our median student is consistently at or above the 55<sup>th</sup> percentile with the year 2005 results at approximately the 62<sup>nd</sup> percentile. (See Figure 4.) Note that the MFT was optional the first two times it was offered.
- Students continue to do better in outcomes 1a, 1c, 1d (areas representing the essence of our program) than outcomes 1b. Our mean results in each of these areas are slightly better than the national averages. (See Figure 5.)
- Student, alumni, faculty and employer surveys have all been satisfactory. (See Figure 6.)

- **Learning Outcome #2:**

Students have had the opportunity to focus in depth on selected areas of computer science.

Analysis:

Students, alumni, and faculty surveys have all been satisfactory. (See Figure 7)

- **Learning Outcome #3:**

Students will have the training to design and implement a large software system and will have the ability to work both individually and collaboratively.

Analysis:

- Faculty evaluation of individual/collaborative skills in CS437 and CS491AB is satisfactory.

➤ Student, alumni, and employer surveys have been satisfactory. (See Figure 8)

• **Learning Outcome #4:**

Students will be able to communicate effectively orally and in written reports.

Analysis:

- Faculty (CS437 and CS491AB) have direct opportunities to evaluate students' oral and written communication skills. Faculty evaluation is satisfactory.
- All students are required to pass a university Writing Proficiency Exam before completion of their requirements.
- Student and alumni surveys are satisfactory. (See Figure 9)

• **Learning Outcome #5:**

Students will have the skills to pursue careers in industry and/or continue their education in graduate programs.

Analysis:

- Alumni, student (seniors), faculty and employer survey results have all been satisfactory. (See Figure 10)
- The self-reported data from alumni suggests that approximately 25% to 30% have received an advanced degree in either Computer Science or a related computing discipline. (See Figure 11)
- Faculty experience indicates that approximately 25% of the graduating class goes on for an M.S. In addition, all "outstanding graduating seniors" from 1990 onwards have completed or are enrolled in an M.S program in a computing discipline.

• **Learning Outcome #6:**

Students will have the skills to adapt to the evolving technologies in computer science.

Analysis:

- Alumni, student (seniors), and employer survey results have all been satisfactory. (See Figure 12)

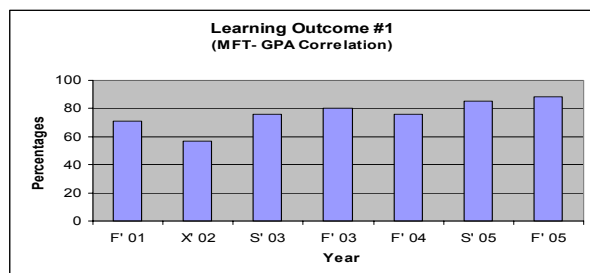


Figure 3

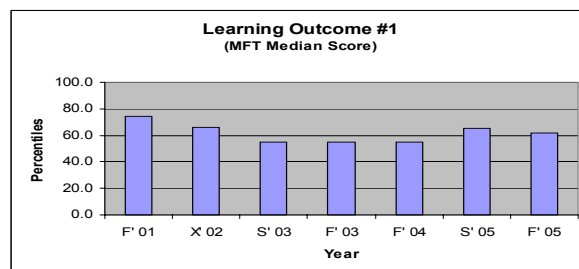


Figure 4

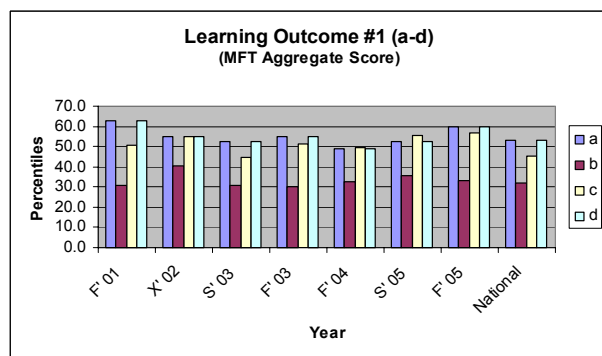


Figure 5

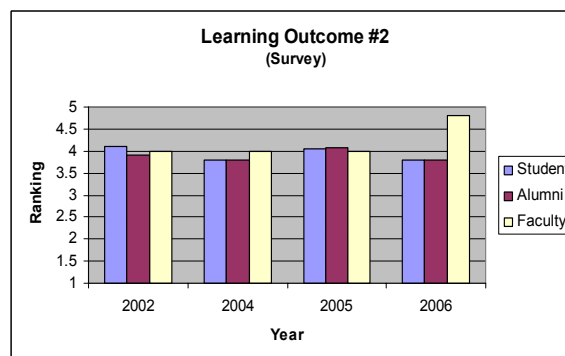
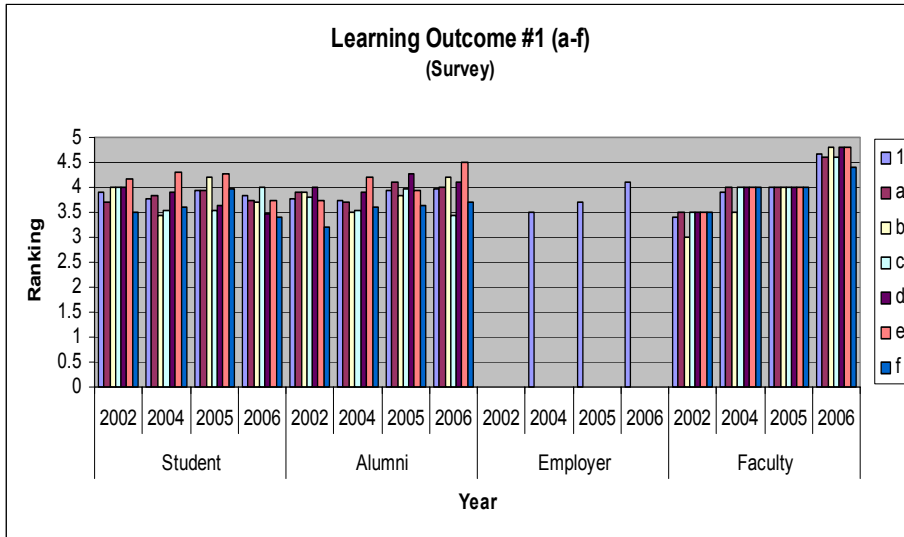
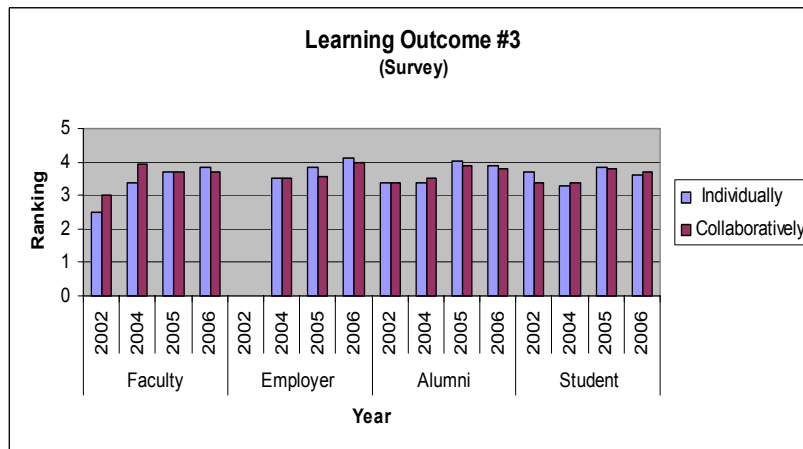


Figure 7



**Figure 6**



**Figure 8**



**Figure 9**

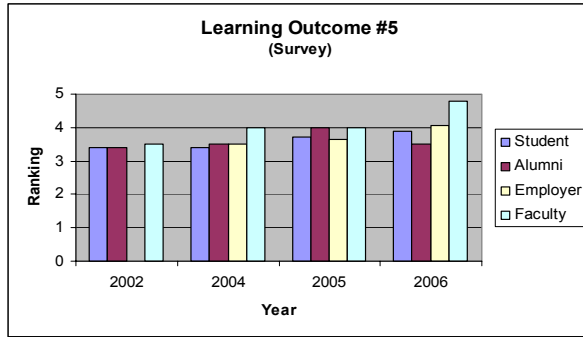


Figure 10

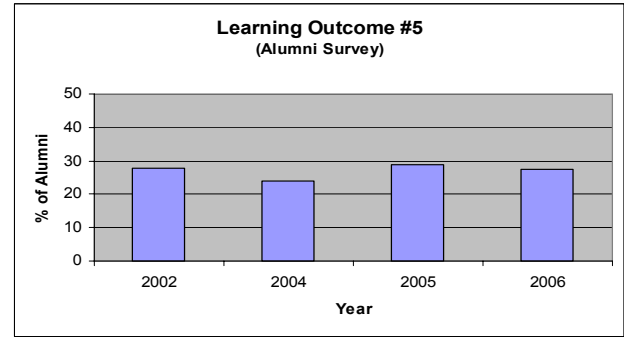


Figure 11

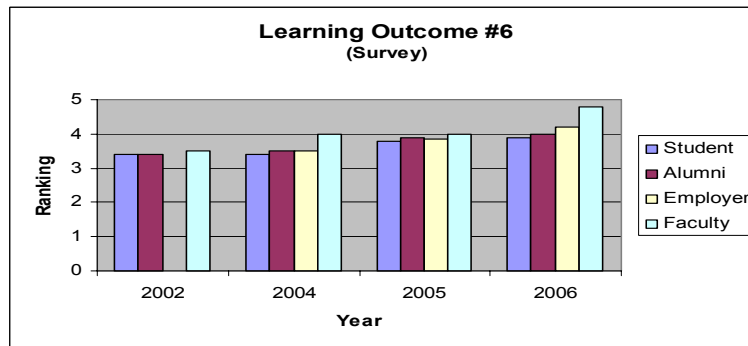


Figure 12

### 3. Program Modifications

Improvement of undergraduate education is the core concept and the guiding principle of the program. Major program changes, which are directly attributable to the assessment results, within the last five years, are as follows:

**2001-2002:** The department instituted a formal assessment process and carried out Loop #2 activities. The department had also completed an external program review in 2000. (University policy is to review each program once every five years.) As a result of these two activities, a number of curricular modifications were proposed. These modifications were motivated by constituent inputs; particularly ABET criteria and ACM curricular guidelines.

➤ Laboratory components were added to the lower division core programming courses (CS201, CS202, CS203). Our existing courses are well defined in content and delivery as 4-unit lecture courses. The additional 1-unit laboratory components provide a stronger foundation for our students and are the best method for delivering this subject matter. Including lab components in our lower division required courses improves our retention rate by helping students as they start out in the program. The lab

facilities provided for the Department make it easier to implement this component. This addition implements Program Review Committee Recommendation 6 and also implements ABET's criteria IV-7. This modification will enhance student achievement of Learning Outcomes # 1.

➤ CS122 was added to the core. Databases play a critical role in almost all areas where computers are used. Relational databases are so widely used that all Computer Science students should know at least the basics of how they work. In addition, knowledge of SQL, which is based on Relational Algebra and Relational Calculus, enhances students' critical thinking skills. This change is supported by faculty assessment and the ACM curricular guidelines. This modification will enhance student achievement of Learning Outcomes # 1.

➤ CS120 and CS320, two courses in web programming were added to the core. According to ACM curricular guidelines, "Web development plays an important programming component the web have become critical foundations of computer science, and it is impossible to imagine that undergraduate programs would not devote significantly more time to this topic." Faculty assessment had

indicated that students need a lower division component (CS120) to introduce this new technology and an upper division component (CS320) to enhance their web development skills. CS 120 covers client-side design and development; CS 320 covers server-side design and development. This modification will enhance student achievement of Learning Outcomes # 1.

- CS245 was added to the core. Networking and operating systems skills are essential for today's programmers. CS245 as a requirement is supported by ABET's criterion IV-8 that requires that students be exposed to a wide variety of systems. Prior to this addition, students were not exposed to as wide a variety of operating systems as we would have liked. This modification will enhance student achievement of Learning Outcomes # 1.
- CS390 was added to the core. As part of our assessment work (feedback from faculty, alumni and industry), we determined that our students were quite weak in large-scale software design. Design is stressed as an ABET requirement in IV-6 and IV-7. CS390 helps students build a stronger foundation in software design. We were also strongly urged by our External Program Evaluator to introduce a software design course in addition to our CS437. This modification will enhance student achievement of Learning Outcomes # 1 and #3.
- CS490 was modified. CS490 was modified to focus strictly on MFT so that students can take the GRE subject test that is required by some graduate schools. Each student is required to solve problems and present their solutions as a way to enhance their communication skills. This modification will enhance student achievement of Learning Outcomes # 4 and # 5.
- A lab component was added to CS437. CS 437 is a team software development course. The lab provides an opportunity for the teams to work together and for the instructor to work with them. This extra lab time makes it possible to require every team to present the group project both orally and in a written format. ABET criterion IV-15 requires that students develop their oral communication and presentation skills. The lecture time saved as a result is now used for in-depth discussions of social and ethical issues in Computer Science as required by ABET IV-17. This modification will enhance student achievement of Learning Outcomes # 3 and # 4.
- CS 444 Computer Architecture was deleted from the program and EE 444. Computer Architecture was added. This change was motivated by the

desire to make use of the expert faculty in the EE Department. This modification will enhance student achievement of Learning Outcomes # 1.

- CS332 replaced CS432 in the core. Faculty assessment determined that our students would be better served by a course that focuses more intensively on the programming paradigms embodied in functional and logic programming languages than by a course that attempts to do a broad survey of many programming languages. The new CS 332 provides that focus. This modification will enhance student achievement of Learning Outcomes # 1.

**2003-2004:** The department carried out Loop #1 activities of the assessment process. Based on assessment data, the Assessment Committee found deficiencies in Learning Outcomes #3 and #4. The Assessment Committee proposed a curricular modification.

- CS390 was modified to be the two-quarter CS491AB sequence. CS390 was originally designed for students to complete an individual project in one quarter. From assessment data, it was apparent that one quarter is not enough for students to complete a significant individual project. Furthermore, interaction among students and between student and instructor is best done in a laboratory environment so that students (a) can learn from each other's presentations and (b) have an opportunity to interact more frequently and more intensely with the instructor. Written and oral communications skills are also better developed in CS491AB. Student self-study skills are also enhanced. This modification will enhance student achievements of Learning Outcomes #3, # 4 and #6.

**2004-2005:** The department carried out Loop #1 activities of the assessment process. No specific deficiencies have been identified from the analysis of Learning Outcomes. The Assessment Committee proposed a curricular modification.

- CS332 was split into CS332F and CS332L; both courses are required for all CS majors. CS332F covers functional programming and CS332L covers logic programming. CS332 had covered two different programming paradigms. Although they contain related elements, the two paradigms are challenging enough and different enough both from each other and from the standard object-oriented paradigm that we determined that they are best taught separately. In addition, faculty who taught this course indicated that students need more laboratory time. A hands-on approach using a programming language to teach

each paradigm—Haskell for functional programming and Prolog for logic programming—makes the paradigms more concrete and easier to grasp. This modification will enhance student achievement of Learning Outcomes #1.

**2005-2006:** The department carried out Loop #1 activities of the assessment process. No specific deficiencies have been identified from the analysis of Learning Outcomes. The Assessment Committee proposed two curricular modifications.

- CS320 is modified to add a laboratory component. This course involves extensive laboratory assignments and is best taught with a laboratory component. Some of the topics previously conducted in a lecture format will be demonstrated by hands-on laboratory exercises. Even though the units are being reduced from 4 units to 3, the amount of time that the course meets is increased by the addition of a laboratory. This extra time justifies converting some of the lecture material to laboratory exercises. This modification will enhance student achievement of Learning Outcomes #1.
- CS301 was added to the core. Topics related to Computer Ethics in the Information Age are covered over a number of courses—especially CS101, CS437, CS491AB. Surveys indicated that this topic had not received satisfactory coverage. It is important for students to understand the responsibilities they bear, to know how their actions can affect both society and individual people, and to appreciate both the good and the harm computers can do. It is also important for students to understand the ethical issues surrounding computers, because computers will continue to play a large part in their lives. This modification will enhance student achievement of Learning Outcomes #1.

#### **Other program changes:**

- New electives have been added which offer more opportunities for the students to focus on selected areas of Computer Science. CS420, CS451, CS480 and CS81 have been added. These additions enhance student achievement of Learning Outcomes #2.

## **4. Conclusions**

The overall goal of our assessment activities is to ensure continuous improvement of the program. Our dedication to assessment has produced the results listed below.

- Fourteen Curricular modifications have been instituted between 2001 and 2006. These have contributed to improvements in each of the Learning Outcomes.
- Results from the MFT have shown improvement.
  - (i) Our median graduating student has performed (at the 55<sup>th</sup> percentile) better than the median (50<sup>th</sup> percentile) of all students taking the exam during 2000-2005. Year 2005 results are significantly better than earlier scores with our median student score at approximately the 62<sup>nd</sup> percentile.
  - (ii) Our mean student score is slightly higher than our median score.
- The correlation between the MFT and student GPAs is greater than 80%. In other words our students seem to do as well in the MFT as they have done in the major. This is a strong validation of our grading system.
- The survey results from students, alumni, faculty and employers have been satisfactory.
- The average satisfaction ratings from student surveys are slightly above 3.5/5 which coincidentally equals the median GPA (2.8/4) of the graduating student. Students have about the same level of satisfaction as their level of attainment in the program.

These results have been produced in the comparatively short five year period during which the department has been in existence. Furthermore, the results indicate that the key to continuous improvement of the program is to have a result oriented assessment process.

## **References**

- [1] Russ Abbott, Vladimir Akis, Valentino Crespi, Huiping Guo, Jiang Guo\*, Elaine Kang, Raj Pamula, Behzad Parviz, Chengyu Sun, "Assessment and Accreditation of Computer Science Undergraduate Program" Proceeding of the 2005 International Conference on Frontier in Education: Computer Science and Computer Engineering (FECS'05), Las Vegas, Nevada, USA, June 20-23, 2005.
- [2] Accreditation Board for Engineering and Technology, <http://www.abet.org/>
- [3] Association of Computing Machinery, <http://www.acm.org>
- [4] Computer Science Department Assessment Website, <http://sun.calstatela.edu/~abet>