Combining Program Assessment with Learning Management for Efficient and Sustainable Accreditation Processes

Chengyu Sun Raj Pamula Russ Abbott

Department of Computer Science
California State University, Los Angeles
csun,rpamula,rabbott@calstatela.edu

Abstract

Academic programs in colleges and universities are often required to conduct periodic program assessment to ensure the quality of the program and to obtain accreditation. A rigorous assessment process is difficult to implement due to the significant effort and resources required. In this paper we present an open-source web-based software system developed at the Computer Science Department at CSULA that tightly integrates program assessment functions with learning management functions. We show that such a system can greatly improve the efficiency to collect, analyze, and present assessment data. Furthermore, building assessment functions into a learning management system, which many faculty and students use on a daily basis, also encourages and facilitates a continuous and sustainable assessment process.

Keywords. learning management system, program assessment, accreditation

1. Introduction

Many colleges and universities conduct periodic reviews of their academic programs to ensure the quality of these programs. In some disciplines, the programs are also required to obtain accreditation to show that the education they provide to the students meets the national or global standard for the discipline.

Conducting a rigorous program assessment can be a daunting task - students need to be interviewed and their portfolios collected, various constituencies need to be surveyed on both educational objectives and learning outcomes, rubrics need to be developed and evaluated, course journals need to be compiled, focus groups need be organized, and the list can go on and on. All the assessment data must be collected, analyzed, documented, and presented in a way that leads to decisions and actions. And what makes it even more difficult is that instead of doing these once every few year, a truly meaningful and effective assessment process must be *continuous* - as stated in `Criterion 4. Continuous Improvement` in ABET Criteria for Accrediting Computing Programs [1] (emphasis mine): "The program must *regularly* use appropriate, documented processes for assessing and evaluating the extent to which the student outcomes are being attained. The results of these evaluations must be systematically utilized as input for the *continuous* improvement of the program."

Many software systems, both commercial [2, 3, 4] and homegrown [5, 6, 7], have been developed to facilitate the program assessment process. These systems, which we refer to as *program assessment systems* (PAS), typically allow users to define learning outcomes, create rubrics for the learning outcomes, and use the rubrics to evaluate student work such as portfolios.

Some PAS, particularly the commercial ones, also support advanced analytics and visualization of the assessment data. The main problem of PAS is that they are rarely used in the day-to-day teaching and learning activities. The common scenario is that a school would employ a PAS to get accreditation, and then the system will not be used again until a few years later when the school is up for accreditation review again. So despite the usefulness of their assessment functions, PAS, in our opinion, actually promote the bad practice of assessment for the sake of assessment rather than assessment for the continuous improvement of the program.

Unlike PAS, learning management systems (LMS) such as Blackboard [8] and Moodle [9] have become part of the daily educational activities in many schools these days. There are two reasons that make LMS a potentially useful tool for program assessment. First, LMS host a wealth of data such as student work that is valuable to program assessment. Second, the use of LMS in day-to-day teaching and learning makes it a suitable place to embed assessment activities for continuous and sustainable assessment processes. However, typical LMS are not designed with program assessment in mind. In particular, they do not organize, analyze, or present data in meaningful ways for program assessment purposes, and their support for common assessment methods is either lacking or limited.

In this paper we present CSNetwork Services (CSNS), an open-source, web-based software system developed at the Computer Science Department at California State University, Los Angeles (CSULA). CSNS integrates the functions of learning management, program assessment, as well as student administration and advisement into one system; and by doing so, CSNS achieves a level of efficiency that is not possible by loosely coupling several special-purposed systems.

CSNS provides a number of functions and tools for program assessment, including:

- Managing program vision and mission statements.
- Managing and mapping of ABET criteria, educational objectives, student learning outcomes, and courses.
- Course-level assessment tools such as rubrics and course journals.
- General-purposed tools that can also be used for program assessment, e.g. surveys, file manager, wiki, and mailing lists.
- Importing and managing data from external standardized tests such as Major Field Tests (MFT).
- Analyzing and presenting assessment data from various sources.

The rest of the paper is organized as follows. Section 2 gives an overview of CSNS, and Section 3 focuses on the functions and tools for program assessment. In Section 4 we discuss our program assessment process and how CSNS helps to facilitate and streamline this process. We summarize our experience and discuss some ongoing work in Section 5.

2. CSNS Overview

CSNetwork Services (CSNS) was originally developed as a simple homework submission system for Computer Science classes in 2006. Over the years, more and more functions were added, and it has become a comprehensive system that supports teaching, learning,

administration, advisement, and program assessment. By the end of 2015, the system is used by 600-1000 users on a daily basis.

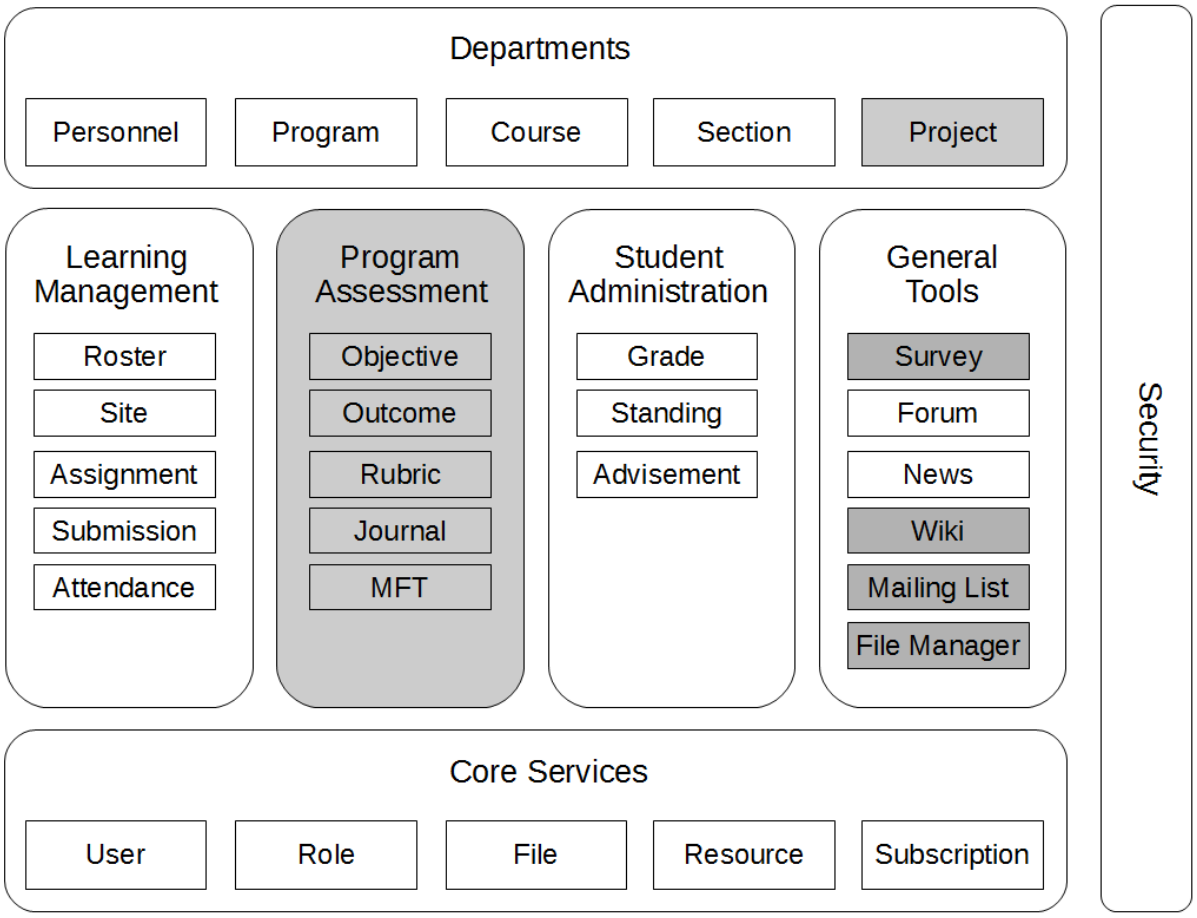
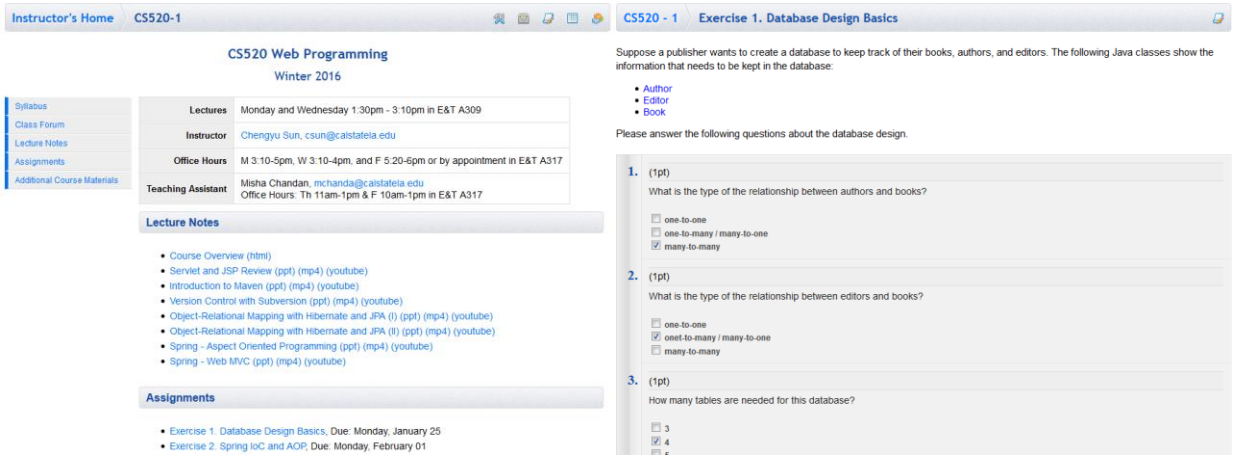


Figure 1: CSNS Overview

Figure 1 shows the architecture of the system with the components used for program assessment highlighted. Although CSNS was originally designed only for the Computer Science Department at CSULA, the current version of CSNS can support any number of departments. Each department can manage their own personnel (department administrators, full-time and part-time faculty), programs, courses, sections, and senior design projects. Each department also has access to a number of functions and tools which can be grouped into four categories: learning management, program assessment, student administration, and general-purposed tools. The core services provide support for users, roles, files, resources, and subscription management. Security is implemented and enforced at all levels.

CSNS provides the basic functions that one would expect from a learning management system. At the beginning of a term, an instructor can create a new class and import the class roster from the university student registration system. A class website can be easily created by either cloning an existing class website or building a new one from a template. As shown in Figure 2(a), a class website consists of one or more *blocks* (e.g. a `Lecture Notes` block and an `Assignments` block), and each block contains one or more *items*, which can be uploaded files, or URL links, or

documents created and edited directly on CSNS. Blocks and items can be easily added, edited, removed (or hidden), and rearranged. An instructor can also post assignments on CSNS, and depending on the assignment type, the students can either upload their solutions as files, or complete the assignment directly online as shown in Figure 2(b). The system can enforce assignment requirements such as due date and the types of the uploaded files, and for online assignments that contain only multiple-choice questions, the system can automatically grade the assignments.



(a) (b)

Figure 2: Class Web Site and Online Assignment in CSNS

CSNS also provides a number of online tools, including surveys, forums, news (i.e. announcements that appear on the front page), wiki, mailing lists, and file hosting. Unlike in other LMS where these tools are tied to classes, in CSNS they can be used both inside and outside classes. For example, an instructor can create surveys for his or her class, and a department chair can create surveys for alumni. And in addition to course forums, CSNS also has department forums for subjects such as job opportunities and advisement, and system-wide forums that discuss issues such as the design and implementation of the system.

The student administration functions in CSNS include keeping track of student grades, changes of academic standings, and advisement records. These functions work closely with other components of the system. For example, changing the academic standing of a student from `B (Undergraduate Student)` to `BG (Graduated with BS/BA Degree)` will automatically unsubscribe the student from the `students` and the `undergrads` mailing lists and subscribe him or her to the `alumni` and `alumni-undergrad` mailing lists.

As shown in Figure 1, CSNS also provides many program assessment functions, which we will discuss in more details in Section 3. Overall, compared to the very class-centric approach taken by most LMS, CSNS is designed to meet a much broader set of needs of an educational institution.

3. Program Assessment Functions

In CSNS, each department can have multiple programs, and each program can have its own vision and mission statement, assessment criteria (e.g. ABET a-k), educational objectives, and student learning outcomes. The system can also keep track of the mappings of the criteria, objectives, outcomes, and courses. As shown in Figure 1, many CSNS components are used for program assessment. In particular,

- *Senior Design Project Listing* hosts the artifacts of each senior design project, including documents such as project requirements document and project report, which are part of a student's web portfolio.
- Program assessment documents are hosted on *Wiki* and *File Manager*, which makes it easier to share and collaborate on these documents.
- *MFT* imports the Major Field Tests scores, compares them to the national distribution, and tabulates and charts the results.
- *Mailing Lists* are used to contact the constituents and get their feedback.

Of all the components involved in program assessment, *Course Journal*, *Rubric*, and *Survey* are particularly important. In this section we cover these three components in more details.

3.1 Course Journal Functions

ABET Accreditation (http://www.abet.org/) requires that a course journal be compiled for each course in the curriculum for accreditation review. A course journal consists of a course description, a course syllabus, all the lecture notes, handouts, assignments and exams, and three samples of student work.

Compiling a course journal used to be tedious work that consumes lots of time and resources. An instructor needs to print out all the course materials, which can easily reach a few hundred pages. And to collect the student samples, an instructor often needs to ask some students to resubmit their assignments, and print out those as well. With CSNS, compiling course journals becomes much easier since most of the course materials and student work (including grades and the instructor's feedback) are already hosted on CSNS. CSNS also provides functions to compile, present, submit, and approve course journals so the whole process can be conducted online (and many trees can be saved).

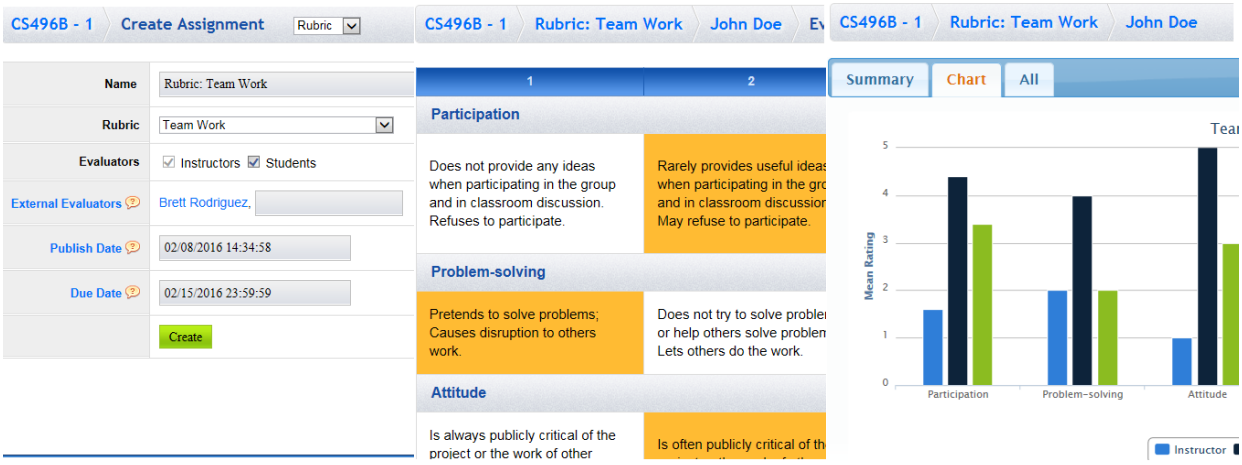
In our ABET accreditation review in 2013, the online course journals were approved by the ABET review team, but there was some confusion from some courses where the course journal items lacked descriptive names or assignments that should be excluded (e.g. make-up tests) were included. After the ABET review, we revamped the course journal functions, and now each course journal item can have a descriptive name and can be a hyperlink, or an uploaded file, or HTML content created directly on CSNS. Also instructors now have the option to exclude some assignments from a course journal.

3.2 Rubric Functions

In the last few years, ABET has put more and more emphasis on the use of rubrics in program assessment. Collecting and processing rubric data can be a complex and tedious process. Most rubrics are evaluated in multiple classes, and rubric evaluation often involves not only the

instructors, but also the students (i.e. peer evaluation) and other stakeholders such as project sponsors and employers. Every year data in the form of thousands of spreadsheets must be collected, processed, and analyzed, which adds significant workload to the already overworked faculty and staff. The Rubrics component of CSNS is designed to simplify rubric management, data collection, aggregation, and analysis.

A *rubric* in CSNS has a number of *performance indicators* evaluated on a numerical rating scale (e.g. 1-5). For each rating for a performance indicator, there is a description about the criteria for the rating. All instructors can create and manage rubrics on CSNS using a web interface. Some rubrics are designated as department rubrics, which can only be edited by department administrators.



(a) (b) (c)

Figure 3: Rubric in CSNS

A rubric can be evaluated in a class as a special "assignment". For example, Figure 3(a) shows a `Team Work` rubric assignment created in a capstone course. Note that there are three types of rubric evaluations: *instructor*, *peer*, and *external*. Instructor Evaluations are conducted by the instructor(s) of the class. Peer Evaluations are conducted by the students in the class. External Evaluations are conducted by external reviewers such as senior design project sponsors, employers, and Industry Advisory Board members.

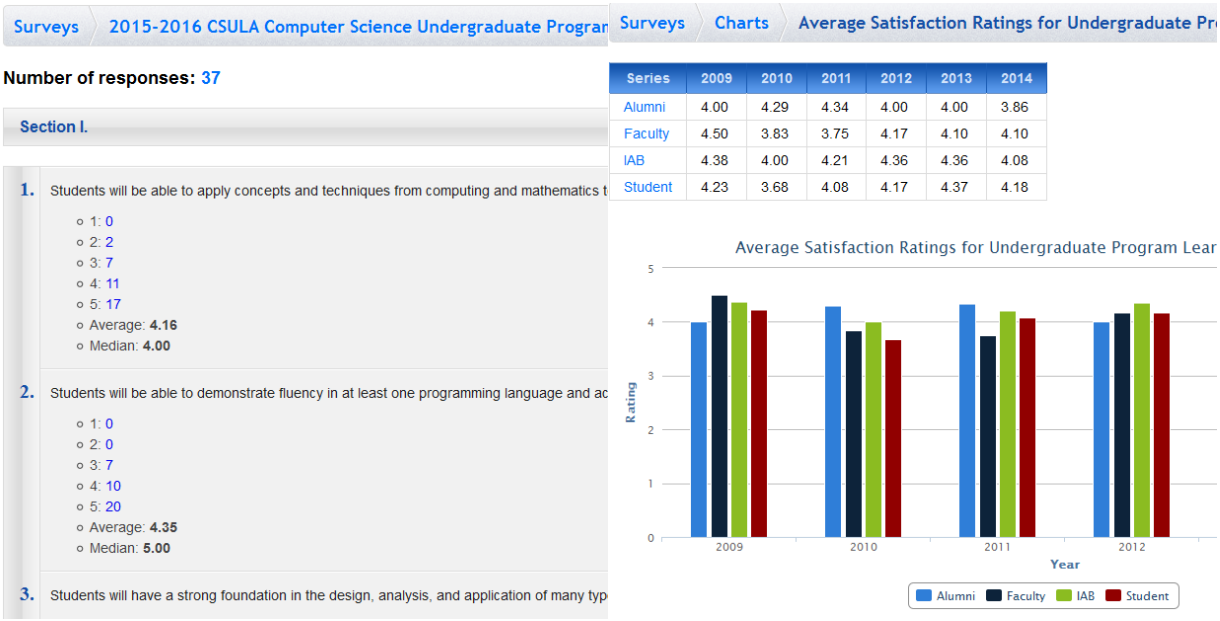
To evaluate a rubric, an evaluator can simply click on the criteria description that corresponds to the rating for a performance indicator, as shown in Figure 3(b). An evaluator can also leave additional comments, which is not captured in the screenshot in Figure 3(b) due to space limitations.

Rubric evaluations are automatically aggregated by the system based on the combinations of several factors: rubric, person, class, evaluation type, and year. The aggregated data is tabulated and charted to help visualizing and analyzing the data. For example, the chart in Figure 3(c) shows the mean rubric evaluation scores for a hypothetical student John Doe by an instructor, an external reviewer, and the other students in the class.

3.3 Survey Functions

Surveys are an important direct measure for program assessment. CSNS supports three types of surveys. An *anonymous* survey is open to the public, i.e. no CSNS account is required, and the system does not keep any information about the people who took the survey. A *named* survey requires the users to log onto CSNS to take the survey, and the system records the identity of the user for each survey response. A *recorded* survey also requires the users to log onto CSNS to take the survey, but the system only records whether a user has taken a survey or not, i.e. a survey response is not connected to a particular user to maintain some level of anonymity.

Each survey has one or more *sections*, and each section may contain a number of questions. Currently CSNS supports three types of questions: *rating questions* that ask the respondents to give an integer rating from a specified range, *choice questions* that ask the respondents to select one or more options, and *text questions* that ask the respondents to enter their answers as text. A survey cloning function is provided to quickly create a new survey from an existing one, and this function is very useful for surveys that need to be conducted periodically (e.g. annual student outcome surveys).



(a) (b)

Figure 4: Surveys in CSNS

One way to summarize and visualize the survey responses is to use the *response summary* function, shown in Figure 4(a). A survey response summary shows the total number of responses and some aggregated data for each question, including the number of responses for each choice or rating, the mean and median ratings for rating questions, and the answers for text questions.

CSNS also provides a *survey chart builder* that allows users to build charts based on the results of different surveys. Specifically, a chart consists of one or more *sequences*, and each data point in a sequence is an aggregated value (e.g. mean or median) of a rating question in a survey. For example, suppose we want to create a chart that shows the satisfaction ratings of the first

learning outcome from 2009 to 2014. The user can create a sequence named `Student`. To add a data point to the sequence, the system will let the user first select a survey, then select a question from the survey. So for the `Student` sequence, the six data points are the first question in each of the six Annual Student Survey on Learning Outcomes from 2009 to 2014. Similarly, the user can add additional sequences based on the surveys to other constituencies such as faculty and alumni. Figure 4(b) shows the resulting chart. Note that new data points can be easily added to the chart after new surveys are conducted.

4. Program Assessment Process

In this section we introduce the program assessment process at the Computer Science Department of the California State University, Los Angeles (CSULA), and show how CSNS helps to facilitate and streamline this process. It should be noted that no matter how good an assessment system is, it only plays an assisting role in the process, and the success of program assessment and accreditation is ultimately determined by the people, not the computers.

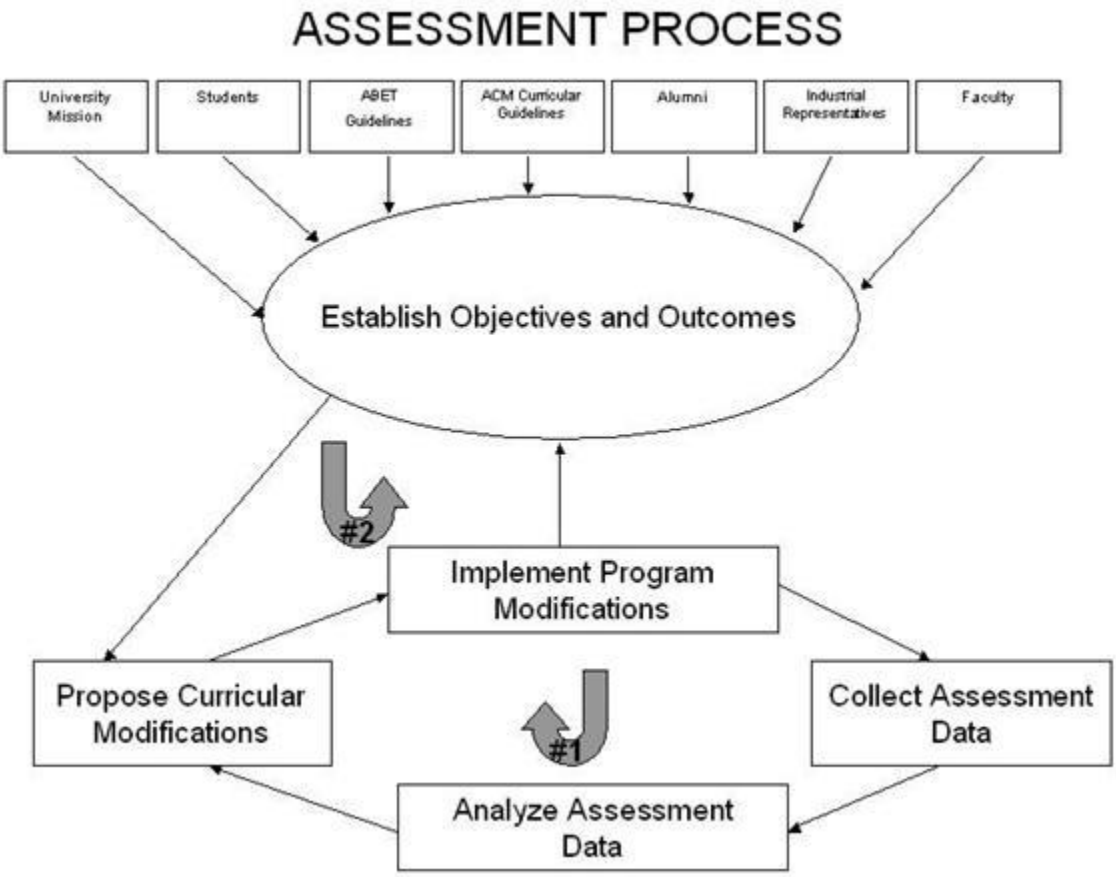


Figure 5: Program Assessment Process at Computer Science Department, CSULA

Figure 5 shows the two-loop program assessment process at the CS department at CSULA. The outer loop, Loop #2, shows the periodic process involved in establishing, assessing, and revising the Program Educational Objectives. This process employs the feedback from various constituencies of the program, and is conducted primarily using CSNS surveys. Loop #2 is a

relatively slow loop where assessment occurs once every five years. Currently we are considering shortening the cycle to three years.

The inner loop, Loop #1, describes how the Student Learning Outcomes are evaluated by collecting data from various measures. Loop #1 is a fast loop where the outcomes are assessed annually to ensure continuous improvement both at the course level and at the program level. Results of each year's assessment measures are used to provide information to help guide refinements to the program.

4.1 Rubrics

We developed six rubrics:

- Team Work
- Oral Communication
- Written Communication
- Software Engineering – Requirements
- Software Engineering – Design
- Software Engineering – Implementation

These rubrics are evaluated on CSNS in the Software Engineering course sequence CS337 and CS437, and the Senior Design Project course sequence CS496A/B/C. These courses are required courses and are offered at least once every academic year.

4.2 Standardized Exams

The Major Field Tests (MFT) is designed by the Educational Testing Service (ETS) to measure the knowledge and understanding obtained by students. The MFT exam is currently utilized by over 230 institutions and more than 9,100 students. This direct measure provides comprehensive data that enables us to evaluate student performance and compare it to programs at similar institutions nationwide. The MFT also provides three indicators:

- Assessment Indicator #1: Programming
- Assessment Indicator #2: Discrete Structures and Algorithms
- Assessment Indicator #3: Systems: Architecture/Operating Systems/Networking/Database

Each indicator provides the average percentage of the correctly answered test questions in that content area for the class as a whole. These indicators are closely tied to our learning outcomes.

Students taking the capstone course CS490 Computer Science Recapitulation are required to take MFT. The scores and the national score distribution data are imported into CSNS, which provides functions to manage, analyze, and visualize the data.

Also conducted in CS490 is an internally designed standardized exam. Modeled after MFT, this exam tests Computer Science knowledge of the students in four areas: Theory, Programming, Algorithms, and Systems. This exam is given as four online assignments that the students complete on CSNS.

4.3 Surveys

Surveys are indirect measures that gather perceptions, opinions, and reflections on learning. Surveys also provide a means to ask qualitative open-ended questions. Every year we survey five of our constituencies - students, faculty, alumni, employers, and industry partners, on the importance and satisfaction with our learning outcomes. The feedback on satisfaction indicates how well we achieve each learning outcome, and the feedback on importance lets us know if we should modify the outcomes themselves.

These surveys are conducted annually on CSNS. We use CSNS mailing lists to contact the students, faculty, and alumni, and we ask our alumni to direct their immediate supervisors to our employer surveys. The surveys of industry partners are conducted at the annual Industrial Advisory Board (IAB) meeting. The survey results are aggregated, tabulated, and charted in CSNS, and serve as the basis for our decision making in continuous improvement.

4.4 Course Assignments

Since courses contribute to the achievement of student learning outcomes, data can be compiled from courses to evaluate those outcomes. These direct measures are in the form of course assignments such as projects, papers, exams, presentations, and portfolios. The assignments we use for assessment purposes are carefully determined by the program assessment committee, and are standardized regardless of the instructors of the course. In most cases, the artifacts collected from these assignments are a part of a student's assessment portfolio.

The learning management functions in CSNS allow instructors to easily create assignments. Depending on the type of the assignment, students can either upload their solutions as files, or complete the assignment online using a web browser. Assessment data is collected from selected assignments every time the course is offered, which is usually once or twice a year.

4.5 Assessment Documentation

We believe a good program assessment process should be clearly defined and well documented. The vision and mission statements, educational objectives, and student learning outcomes of our program are available online to the public. We also systematically document our assessment process, including the mappings of ABET criteria, objectives, outcomes, and courses, as well as the assessment measures used and data collected for each learning outcome. Furthermore, all the assessment documents, including not only the self-study report and annual assessment reports, but also assessment-related plans, papers, presentations, meeting agendas, and so on, are also available online to various parties involved in the assessment process. All the assessment documentation is hosted on CSNS File Manager and Wiki, which provide easy and controlled access, and facilitate sharing and collaboration on the documents.

5. Conclusion and Ongoing Work

In this paper we introduce CSNetwork Services (CSNS), an open-source, web-based software system that combines the functions of learning management, student administration, and program assessment. We show that such a system can greatly reduce the time and resources required to collect, analyze, and present assessment data, so the institution can focus more on perfecting the assessment process and improving teaching and learning. Furthermore, building

assessment functions into a learning management system, which many faculty and students use on a daily basis, also encourages and facilitates a continuous and sustainable assessment process.

We have been continuing to improve CSNS since it went into operation in 2006. The current work is focused on the organization and presentation of the assessment data in the system. In particular, we are working on a user interface that provides both program assessors and reviewers easy access to all the assessment methods, results and analysis in an intuitive and coherent manner.

References

[1] ABET Computing Accreditation Commission. Criteria for Accrediting Computing Programs, 2016-2017, October 2015.

[2] Taskstream. https://www1.taskstream.com.

[3] LiveT ext. https://www.livetext.com.

[4] AEFIS. https://www.aefis.com.

[5] Aldo A. Ferri and Wayne E. Whiteman. A web-based tool for course-level assessment of student learning outcomes. 2015 ABET Symposium, April 2015.

[6] Wael Hosny Fouad Aly. Towards improving the assessment process experience. 2015 ABET Symposium, April 2015.

[7] Kevin Ayers, Adam Fontecchio, and Kristin Imhoff. Using a course assessment materials database for a multi-program visit and how it assists preparation for accreditation. 2015 ABET Symposium, April 2015.

[8] Blackboard. http://www.blackboard.com.

[9] Moodle. https://moodle.org.