

ACAT: A Web-based Software Tool to Facilitate Course Assessment for ABET Accreditation

Eugene Essa, Andrew Dittrich, Sergiu Dascalu, Frederick C. Harris, Jr.

Department of Computer Science and Engineering

University of Nevada, Reno

Reno, NV USA

essa, dittricha, dascalus, fredh@cse.unr.edu

Abstract—There are several institutions that accredit educational programs and require documentation to ensure that an educational program regularly meets certain criteria. This paper focuses on ABET program. They require that programs show student achievement and certain course outcomes. Documentation of this requirement is particularly burdensome. There is no standard method of generating these reports, so each institution handles it differently. This might involve manual collection of the data which is time very consuming. A software tool that facilitates this collection of the data and automatically generates the required reports would save institutions time and money. This paper presents such a tool, named ACAT (ABET Course Assessment Tool), it is a web-based application designed to assist in the collecting of data and generation of standardized assessment reports. This paper focuses on design and usability aspects of the proposed ACAT tool and provides implementation and operation details.

Keywords-Accreditation, User interface development, Assessment, Courseware, Outcome-based learning, Web interface

I. INTRODUCTION

Achieving an accredited status indicates that the program meets certain elite standards set by the accrediting institution. This indicates that the program meets certain standards set by the accrediting institution, which typically consists of members of the academic community, professionals, practitioners, and governing boards [1].

This accredited status makes the school a better choice for students, because it indicates that the students will gain the knowledge and skills necessary to be a productive member of their chosen profession [2]. Engineering programs are typically accredited by ABET. They define a set of criteria that an engineering educational program must meet in order to be accredited. The process to achieve and maintain an accredited status involves generating reports documenting that the school were successful in achieving these criteria. Often department personnel are presented with an undue and unnecessary burden in preparing this documentation [3].

One especially burdensome part of accreditation is ABET Criterion 3. It is defines as set of 11 course outcomes, (a) though (k), each of which identifies a specific ability or skill that a student must achieve upon graduation from an

accredited program. ABET requires that an educational program have a system of assessment in place in order to periodically document the student's achievement of these outcomes [2].

Determining which course outcomes are assessed for each course is the first step. Once they are defined, then the degree to which the students achieve these outcomes can be measured quantitatively through scores earned on assignments, quizzes and exams. Either these scores can be used to determine if students are achieving the outcome or an assessment team reviews the student's work and rates their achievement of the course outcome.

There are also indirect assessment methods which may include course evaluation surveys, alumni surveys, or job placement statistics [4]. A typical course assessment will contain a combination of direct and indirect methods [5]. This data is collected, normalized and averaged for each course outcome, and the result indicates the student's achievement

The process is very time consuming and tedious. It requires storing a large amount of data, including descriptions of the instruments of direct assessment, individual scores on each instrument of direct assessment, and other supplementary data such as survey results and documentation of course changes. Preparing the reports requires analyzing the data, generating tables and graphs, which must be presented in a consistently formatted document. With different teaching styles and organizations of course data, even in the same institution, inconsistencies can often results. There is no standardization method for this assessment process.

In this paper, the ABET Course Assessment Tool, or ACAT is presented. The goal of this tool is to simplify the assessment process for the faculty involved, while placing minimum requirements of changes to their existing teaching methods. It is focused on helping collect and organize the results of the instruments of direct and indirect assessment, analyzing this data, and generating standardized reports that can be used for the accreditation process.

This paper, in its remaining part, is organized as follows: Section 2 surveys related work aimed at facilitating the assessment process; Section 3 describes the requirements and

guidelines followed for building our ACAT web application; Section 4 details ACAT's design, including system design, database design, and user interface design; and Section 5 finalizes the paper with directions of future work and conclusions.

II. RELATED WORK

There have been many efforts to streamline and facilitate the assessment process. In DeLyser and Hamstad [6], the faculty was able to reduce the assessment workload significantly by eliminating redundancies in the assessment process. This was done by carefully selecting which instruments of direct assessment were included in the assessment process in order to prevent assessing the same student twice for the same course outcome. Blandford and Hwang [7] suggest using sampling to reduce the overall workload. This can be done by using only a subset of the instruments of direct assessment, by using a subset of the students, or a combination of both. Yamayee, Albright, Inan, Kennedy, Khan, and Murty [8] placed an emphasis on creating instruments of direct assessment that were focused on a particular course outcome and easy to evaluate. These improvements are helpful in reducing the amount of work overall, and were shown to be effective in streamlining the assessment process, but they don't eliminate the need to collect and analyze the assessment data, and to prepare assessment reports.

Some programs have adopted courseware to facilitate the assessment process. Booth [9] describes a database design that could be used to organize the data required for an assessment report. This database mapped assignments to course outcomes, collected data for each assignment to measure achievement of the course outcomes. It also collected artifacts of student's work, and documented changes to the course. This database is an effective method for organizing the information required for an assessment report, and allowing easy access to that information when preparing reports.

Abunawass, Lloyd, and Rudolph [10] describe how the University of West Georgia switched from WebCT to an open source course management software called Moodle, and were able to adapt this software to store student portfolios. These student portfolios were then used as the basis for the assessment process. This was a major improvement over their existing assessment process, and helped to manage and store the vast amount of data required to document the student's achievement of course outcomes. However, this requires a dramatic change in the organization of all course data, which may not be feasible at all institutions. Both of these cases demonstrate that a software solution can be effective in streamlining and automating the assessment process.

At our institution, the existing assessment process is as follows. Define the course outcomes for each course.

Define the instruments of direct assessment that measure the student's achievement of those course outcomes. Collect the students' scores for each of the instruments of direct assessment and normalize them to the same scale, for example 0 – 5. Calculate the average of the students' scores for the instruments of direct assessment for each course outcome. Collect the results from a student self-assessment survey, in which the students give their opinions on their achievement of the course outcomes.

Once all of this data is collected, it can be entered into an assessment report template. This is a Microsoft Word document that contains tables and charts that show the achievement of the course outcomes. This process requires manual data entry and manual formatting of the resulting report, and is very tedious and time consuming, and certainly in need of vast improvement.

To address this, we have developed the ACAT software, aimed at making the assessment process much more effective and user friendly. The key components of this web-based tool are a rich user interface, its capability to generate the required assessment reports, a fully integrated database, and accessibility from any computer connected to the internet.

III. REQUIREMENTS AND GUIDELINES FOR ACAT

The input data for the ACAT tool includes exam questions, homework assignments, students' scores and grades, and student self-assessment survey results. This data is highly sensitive, and must be restricted to only approved individuals. This requires that access to this tool and the reports generated using this tool be restricted as well. This can be accomplished by requiring a username and password to log in, and using encrypted network connections such as SSL. This also requires an administrator for this software who can be trusted to enforce policies and procedures to keep this sensitive data private.

One input to the tool is students' scores on the instruments of direct assessment. Each educational program has different methods of tracking this information. This method varies from professor to professor within the institution. It would be ideal if the input data format could be standardized and stored in a file which could be uploaded into the tool. However, it is not feasible to force professors to change the way they keep track of scores just to accommodate this tool. It is also not feasible to require manual entry of each score for each student. This would not make the tool user friendly. The most efficient method to input data is using copy-and-paste.

In order to make the data entry as easy as possible for the user without forcing a specific format for storing student scores, support should be added to allow copying and pasting from this spreadsheet into ACAT.

Not every student is included in the final report. Students that do not finish the course should not be included in the final report. It may also be desirable to perform a random

sampling of students for large classes. Removing some students from the report could be done manually before entering the data, but it would be more user-friendly to add this feature to the tool.

The course outcomes that are measured will change. The accrediting institution may change the requirements from time to time, forcing the educational organizations to change their reports. The educational organization may add their own custom course outcomes in order to measure outcomes not covered by the accrediting institution. In order to accommodate these changes, the tool should allow the user to change the course outcomes. The previous course outcomes should be preserved to allow reviewing reports from the past. This can be accomplished by defining a set of course outcomes, and selecting which set of course outcomes should be used for a particular assessment report.

The course outcomes provided by the accrediting institution have a default description. These default descriptions are a general definition of the course outcome and are meant to cover a wide variety of courses. A user may need to customize this description for a particular course. For example, ABET's description for course outcome (g) is "an ability to communicate effectively" [2].

This may be custom tailored for a software engineering course by restating it as an ability to communicate the scope, specification, and status of a software project effectively through scope and vision documents, software requirements specification documents, design documents, and project status reports. To accommodate this, the tool should allow the user to create a customized description for each of the course outcomes for a specific course.

IV. DESIGN

Using the information gathered from interviews with stakeholders and users, an analysis of the existing system, and research into the assessment processes in place at other universities, the design for ACAT was developed. The following sub-section discusses some considerations for the design of ACAT, while the remaining ones present details of ACAT's system design, database design, and user interface design.

A. Design Considerations

Interactive web pages using standard form components allow for a standardized and user-friendly interface. The ACAT system is divided into four distinct components: *authentication*, *data entry*, *report generation*, and *administration*.

Users of the system can be categorized as either general users or administrators. *Authentication* occurs with users logging into the system with a user ID and a password. Since ACAT is designed for a specific purpose, only designated users have access to its features and functionality. This requires an administrator to register individuals prior to their first logging in. Additional functionality for resetting passwords and forgotten passwords is also needed. Some personal information such

as name, title, and email may also be required. ACAT also provides facility to enter and edit this data and finally the ability for all users to log out.

The *data entry* portion of ACAT comprises the majority of the user interface web pages. These pages require flexibility, usability, and efficiency to ensure the effectiveness of the system and to justify its development. Users are able to begin a new report or modify an existing report. Forms are available to enter or edit the following: course, semester, whether the course was taught previously, prerequisites, whether the students were prepared by the prerequisites, changes made, future changes, student comments, number of students per discipline, course outcomes, instruments of direct assessment, student self assessment scores, and outcome averages.

The system provides *navigation* to allow easy access to any of the data entry pages. This allows for direct access to any data that needs to be immediately modified. For new assessments, some information is required before other parts can be entered. ACAT ensures these requirements are met and dynamically shows additional pages for data entry as the assessment is being completed.

Once the required information has been entered into the system, a report can be generated. The actual *generation* of the report is performed by the system without user interaction. The report is presented in a common document format, PDF. The system automatically calculates the average outcomes for the scores that were input. The report also includes tables and graphs per the report samples and stakeholder requests.

The final component is for *administrative* use. This area is only accessible to users designated as administrators. The functionality includes the ability to create new users and administrators, reset passwords, edit personal information, and manage course outcomes.

B. System Design

As a web based system, HTML is used to display pages to the user. This HTML is dynamically generated using PHP server side scripting. The scripts generate the pages using data stored in a MySQL database. The entire system can be viewed in layers, as shown in Fig 1. At the top level is the *presentation layer*. This contains the GUI subsystem, which uses HTML to display data to the end user. Interface to provide for data input are also located at this layer.

Below this is the *business logic layer*. This layer contains the General Pages subsystem, the User Pages subsystem, and the Administrative Pages subsystem. The General Pages subsystem implements functionality for logging in, logging out, displaying the home page, and other basic functionality. The User Pages subsystem implements all functionality for entering assessment data and generating reports. Authentication subsystem controls access to different pages within the website based on the user's credentials.

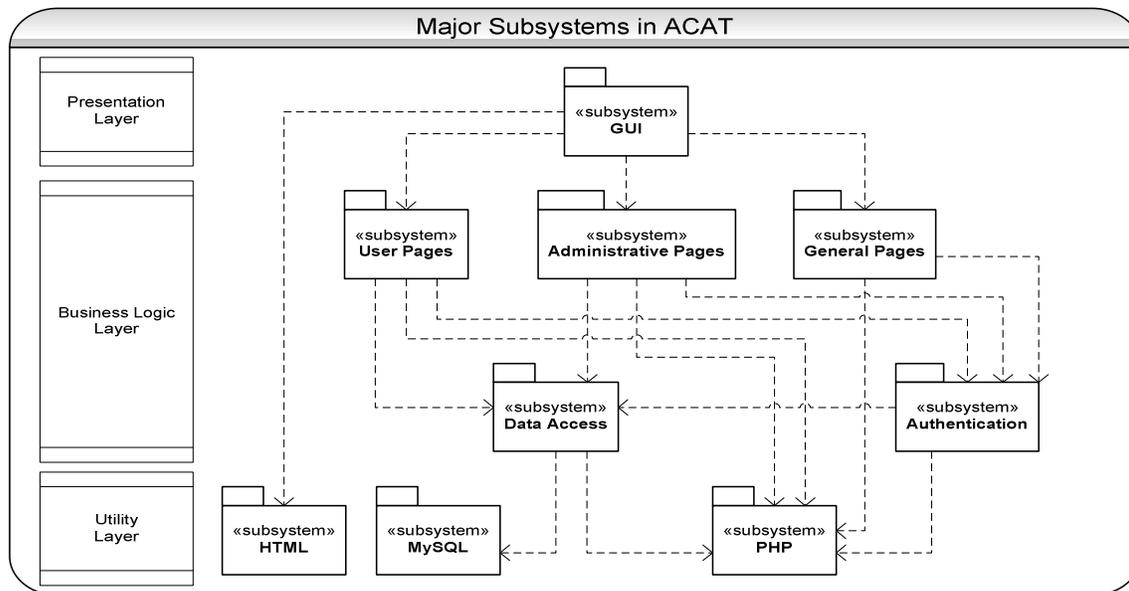


Figure 1. Major subsystems in ACAT.

At the bottom is the *utility layer*, which includes the HTML subsystem, the PHP subsystem, and the MySQL subsystem.

C. Database Design

The database design represents a normalized database schema that stores all the data items needed to produce all of the required sections of the assessment. Flexibility and scalability were primary concerns in the design to allow for smooth implementation of current requirements as well as future modifications. While the current implementation is with MySQL, the database design is generic to allow for implementation on any database platform.

Data validation from the forms occurs at the client level as well as the server level. Client side scripting allows for validation to occur before the form is sent and processed on the server. This provides immediate feedback to the user without having to wait for a cycle from the server as well as the time for the page to reset in the browser. Additional server side validation ensures clean data is stored in the database.

D. User Interface Design

One of the primary goals of the ACAT system is to provide a visually attractive and highly usable interface for the user. This is accomplished through standard web controls and style sheets to ensure continuity throughout the site.

The user interface web pages are presented as standard web forms. This common and recognizable interface allows for a high amount of intuitive usability. Web form controls such as text boxes, radio buttons, and check boxes represent data input facilities that are well known to most of the experienced computer users.

Users are authenticated through a standard log-in process of providing an ID and password. Users are then directed to a welcome page that displays existing assessments and also a link for creating a new assessment. The left panel contains navigation choices and is always available. The design and layout of this page is shown in Fig. 2.

Pages for entering data are accessed from the assessment summary page. Any previously entered data is also displayed on this page. Links to add or edit information are available. Pages such as the one depicted in Fig. 3 are used to input or edit data.

The forms are displayed in tabular format with the labels in line. Error messages are displayed next to the field containing the error, as shown in Fig. 3, and are displayed in red. Controls to submit or reset the form are displayed as links instead of as standard buttons to facilitate server side validation.

The reports can be generated from a link on the summary page. The assessment information is displayed in graphs and tables providing a succinct summary of the ABET Criterion 3 outcomes for each course.

These reports are generated using an open source PDF generation program, TCPDF. This flexible and easy to use friendly PHP class provides functionality for creating PDF documents in the browser. Using the PDF format for reports will allow for easy printing, saving, and distribution.

Examples of generated reports are shown in Fig. 4 and Fig. 5. Due to space limitations only a few of ACAT's features and functions have been illustrated in this section. However, we hope they can provide a good idea of the tool's capabilities and utility.

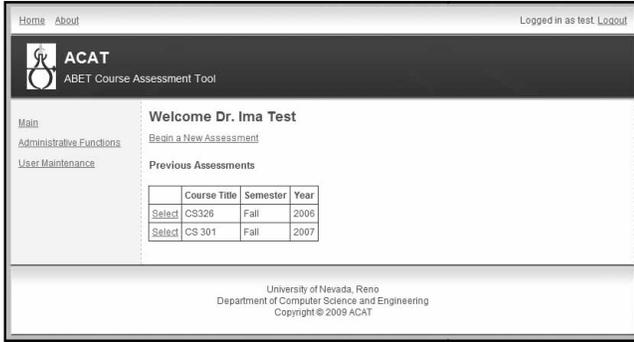


Figure 2 The ACAT welcome page. Navigation on the left panel has options for administrative functions and user profile maintenance.

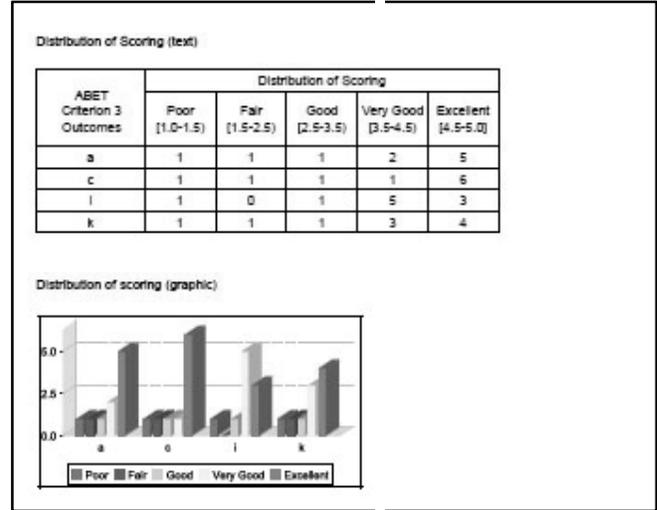


Figure 5 Distribution of scoring shown in graphical and text formats on the assessment form.

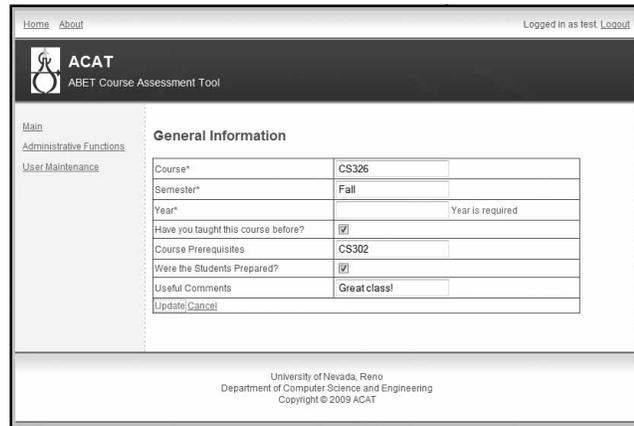


Figure 3 Form for editing general information. An error message is displayed adjacent to the field creating the error.

V. CONCLUSIONS AND FUTURE WORK

A need exists to automate and standardize the process for generated ABET course assessments. A detailed analysis and requirements development was performed and a comprehensive design was created for developing a software tool, ACAT, aimed at facilitating the assessment process. The emphasis on the design centered on ease of use and functionality.

Based on this design, a prototype of ACAT was developed. Although it is not fully feature complete, it has the most important functionality working. It provides the facility for user to enter all of the data for a course assessment, including general information, course outcomes to be assessed, instruments of direct assessment for each outcome, student scores for each instrument, and student self evaluation survey results. The system has been used in several trial cases, and shown to be a viable tool in data collection and reporting.

The next step in the evaluation is to deploy the fully functioning system to allow for stakeholder testing. Once approved, a usability study will be conducted with the staff that will use the system. Metrics discussed in this paper will be used to evaluate the effectiveness of the tool.

ACAT also has the potential to support the assessment process in other accredited universities. ACAT collects data that is common in assessment reports for, and could be tailored to a specific program's needs with slight modification. For example, the University of Michigan at Dearborn prepares assessment reports that list the course outcomes being assessed, what the students were asked to show their achievement of that outcome, and the average score [12]. ACAT collects this data currently, and could be adapted to prepare reports in the desired format for this program as well as many other similar programs.

Achievement of Objectives/Demonstration of Outcomes

Did the students demonstrate achievement of the learning objectives/expected outcomes specific to this course? In the table provided, cite evidence using student responses on the student self assessment questions and evidence from your direct assessment of student work.

If sampling, please indicate the approximate percent of the class sampled:

Computer Engineering

Mapping between ABET Outcomes - Course Outcomes - Instruments of Assessment

ABET Criterion 3 Outcomes	Student Self-Assessment	Course Outcomes	Instruments of Direct Assessment by Instructor	Outcome Average
a	4.168	My custom definition	Midterm - pb. 6	4.125
c	4.333	an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability	Final - pb. 8	4.250
l	4.450	a recognition of the need for, and an ability to engage in lifelong learning	Midterm - pb. 9 Final - pb. 11	3.625
k	4.000	an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice	Midterm - pb. 3 Final - pb. 4	3.500

Figure 4 Portion of the ACAT generated assessment report mapping the course outcomes and instruments of direct assessment to the ABET outcomes.

REFERENCES

- [1] The Council for Higher Education Accreditation website at <http://www.chea.org/>, accessed May 9, 2009.
- [2] ABET official website at <http://www.abet.org>, accessed Oct 27, 2008.
- [3] J. Enderle, J. Gassert, S. Blanchard, P. King, D. Beasley, P. Hale, & D. Aldridge, The ABCs of preparing for ABET, *IEEE Engineering in Medicine and Biology Magazine*, 22(4), 2003, 122-132.
- [4] Z. Yamayee, R. Albright, M. Inan, M. Kennedy, K. Khan, & V. Murty, Work in Progress – Streamlining Assessment Process in Response to a Successful ABET Visit, *Proceedings 35th Annual Conference, Frontiers in Education*, Indianapolis, IN, 2005, 19-22.
- [5] K. Sanders, & R. McCartney, Program Assessment Tools in Computer Science: A Report from the Trenches, *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, Reno, NV 2003, 31-35.
- [6] R. DeLeyser & M. Hamstad, Outcomes based assessment and a successful ABET 2000 accreditation at the University of Denver, *Proceedings of the 30th Annual Frontiers in Education - Volume 01*, 2000, T1A/1-T1A/6.
- [7] D. Blandford & D. Hwang, Five easy but effective assessment methods, *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, Reno, NV, 2003, 41-44.
- [8] Z. Yamayee, R. Albright, M. Inan, M. Kennedy, K. Khan, & V. Murty, Work in progress – streamlining assessment process in response to a successful ABET visit, *Proceedings 35th Annual Conference, Frontiers in Education*, Indianapolis, IN, 2005, 19-22.
- [9] L. Booth, A database to promote continuous program improvement, *Proceedings of the 7th conference on Information Technology Education*, Minneapolis, MN, 2006, 83-88.
- [10] A. Abunawass, W. Lloyd, & E. Rudolph, COMPASS: a CS program assessment project, *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, Leeds, United Kingdom, 2004, 127-131.
- [11] I. Sommerville, *Software engineering* (Essex, England: Addison Wesley, 2007).
- [12] B. Maxim, Closing the loop: assessment and accreditation, *Journal of Computing Sciences in Colleges*, 20, (1), 2004, 7-18.