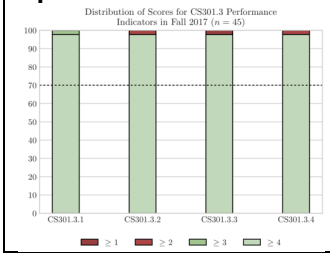
CS486C – Senior Capstone Design in Computer Science
Project Description

Project Title: Graphical User Interface for Program Accreditation Data

Sponsor Information:



Dr. Toby Hocking

Assistant Professor
School of Informatics, Computing, and Cyber Systems
Machine Learning Research Laboratory
Email: toby.hocking@nau.edu
Phone: (928) 523-5209

Project Overview:

Program Accreditation is the primary mechanism for quality control in higher education, ensuring that educational programs have clearly identified what learning outcomes they are trying to achieve, are effectively measuring achievement of those outcomes, and are using the results of this analysis to implement rational, data-driven program improvements. Accreditation is particularly important in the engineering disciplines, as there is a vital need to ensure that graduates leaving a program with an engineering degree actually have the necessary skills to practice as safe and effective engineers.

The primary accrediting body for engineering programs worldwide is known as ABET (the Accrediting Board for Engineering and Technology), with specific sub-divisions dedicated to accreditation of each of the various engineering programs. The overall approach/process that ABET uses is common across all of its divisions: an engineering program that would like ABET accreditation must develop a set of "Program Objectives", which are essentially the macro-scale things that it aims for graduates to be able to do with their degrees. Each of these Objectives is then tied to a set of "**Program Learning Outcomes (PLOs)**", which are the specific things that students would have to learn over the course of their degree program in order to achieve each Program Objective. The department offering the degree program must then explain, for each PLO: how and where it will measure achievement of the PLO, how it's going to review the data being collected to see how well the PLO is being achieved, and how it then "closes the loop", by changing/improving their degree program to address learning deficiencies revealed. All of this process fits into a six-year "accreditation cycle", meaning that a program must undergo extensive review (like an audit) every six years to get re-accredited.

From this overview, it should be clear that managing the ABET process – collecting the student performance data you promised to, and (especially) monitoring it to keep an eye on student achievement of the learning outcomes – is an arduous task…and yet, if you don't do it right, your entire program accreditation is on the line.

The Project:

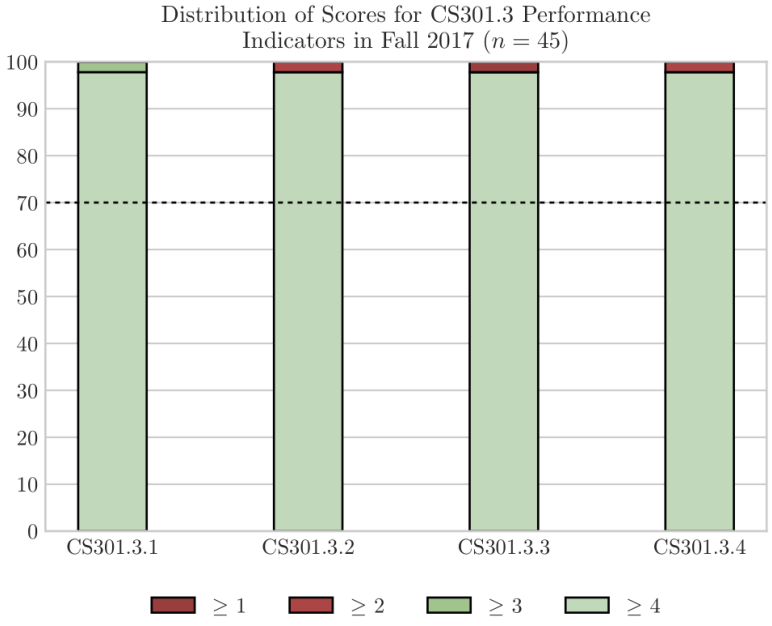
This project is about creating a simple tool to help visually summarize ABET data collected by an ABET-accredited degree program; more specifically, its main goal is create a tool for visualizing the specific data collected for the CS program here at NAU…but with an eye towards making it configurable enough to be used by other programs with their own data formats.

The way that ABET data collection is configured for the NAU CS program is fairly simple. As outlined above, we have established a set of PLOs (program learning outcomes) that we must achieve for all graduates. For example, **ABET Outcome 3** is the ability to "communicate effectively in a variety of professional contexts". Achievement of this PLO is measured in one or (more often) several courses through “performance indicators” that are customized to each such course. For instance, achievement of our Outcome 3 is assessed in CS301 (Ethics) using the following performance indicators:

- **CS301-3-1:** Flaws and errors in spelling, grammar, punctuation, syntax, and diction
- **CS301-3-2:** Effective communication of document purpose, motivation, and organization.
- **CS301-3-3:** Effective organization of ideas to create a coherent, compelling argument. 'Depth, Quality of Content'
- **CS301-3-4:** Completeness: All key topics covered at the appropriate level of detail. 'Breadth, Scope of Content'

Each of these four indicators is evaluated for each student in each offering of CS301 by way of a “rubric”: The course instructor is given a scale from 1-5, with a description of what kinds of evidence or skill level is associated with each level. At the end of the course, the instructor then rates each student on this scale, based on tests, homeworks, in-class interactions, and other deliverables. The data generated in this way is archived and managed is quite simple: For each course, there is a spreadsheet for each outcome measured by that course. The spreadsheet has “tabs” in it: The first tabs have the list performance indicators being measured in the class and the evaluation rubrics; then there is a tab for each course offering, with a list of all students in the class that offering, and their scores on each of the indicators. So each offering, a new tab is added to the appropriate course spreadsheet, and the outcomes measures for those students are entered there by the course instructor. These tabs have formulas at the bottom of the columns that indicate stats: how many students score above 3 (“passing”) on each indicators, high score, low score, percentage achieving goals, etc.

This spreadsheet-based method has proven to be efficient and effective, but it has a problem. Although the spreadsheets make it (relatively) easy to see the performance on the indicators *for a particular offering*, it’s lots harder to understand the performance ***over time, across offerings***…but it is precisely this “trend” that is most important in assessing PLO performance within the six-year period. Has performance on a given PLO generally improved, stayed flat, or (oh no!) declined over the course of the six-year accreditation window. Accreditors will want to see “continuous improvement” in the program learning outcomes!



The aim of this project is to make a simple, GUI-based tool that allows various people concerned about the accreditation process (class instructors, department Chairs, Deans, etc.) to easily and instantly review progress on the degree program’s PLOs. An authorized party should be able to authenticate, and then see a dashboard of the degree program’s PLOs, each with some high level

indicator (e.g. red, orange, green) of current achievement status. The user could then drill down into a given PLO to see the courses in which that PLO is being evaluated. Looking into a given course would list the performance indicators used for that course, as well as a graph showing achievement on those indicators over time. In addition, it would list the Excel spreadsheet it is using as a data source, and the years/semesters for which it has ABET data, i.e., it indicates what data are being used to generate the graph. We envision this as a web app, but the team can propose an alternative if appropriate. Some specific highlights and feature levels include:

Phase 0: Minimum Viable Product

- Ability to somehow direct or configure the tool to the indicate the directory holding the Excel sheets containing the ABET data; these are currently stored in a secure Dropbox folder accessible only to CS faculty. The team may assume a rigid naming scheme for files to make it easy to identify which Excel file holds data for which course/PLO (e.g. SICCS/Assessment/Courses contains files like "EE 310 (2, 6).xlsx" and "CS 421 (6) (A, K).xlsx")
- Ability to list the Excel sheets in the specified directory, and to "open" one with the tool.
- Ability to simply graph the performance across all indicators for a single semester (i.e. for one tab in the spreadsheet); this would look like the graph in the figure above.
- Ability to extract the appropriate data from the Excel sheet, and to generate a graph *over all semesters/offering for which there are tabs in the spreadsheet*. This gives the critically-needed "performance trend over time visualization" mentioned above. The graph would be returned as an image file or, ideally, would be embedded in an html page generated by the app. Users could then view the page (and its graph) using a standard browser.
- Table visualization overview with outcomes on one axis and courses on the other. Clicking on entries of the table zooms to relevant details (another data viz).

Phase 1: A nicely useful GUI tool

- Allows users to authenticate in some way. Ideally using the NAUauth framework (provided by ITS) to allow faculty to use their NAU credentials to log in.
- Configurable to point the tool to a directory holding all ABET data.
- Provides a "DashBoard" page that lists all PLOs found (based on looking at the Excel spreadsheets present) in the designated directory. For each PLO, it lists the courses where it is being measured and, for each such course, gives a simple indicator of current achievement status.
- User can click to examine a particular course, its performance indicators, and their trend over time as a graph. This is based on instantaneous extraction and graphing of the data in the Excel file. Below, it also lists all semesters included in the analysis in a table, along with the summary (average, min, max, pct achieving, pct failing) values for each semester.
- Ability to adjust the "target achievement level". Normally we shoot for a score of 3 or better…but one might want to see how altering target achievement level affects the stats.
- Ability to somehow export the graphs as image files, e.g., for inclusion in an ABET report.

Phase 2: Stretch goals

- Ability to adjust the data range being graphed dynamically in the graphing tool. By default, it graphs all semesters/offering for which there is data in the spreadsheet, but there are controls on the graph viewer to include/exclude certain semesters/offerings or otherwise adjust time range shown.
- Ability to "de-aggregate" the data to selectively investigate subgroups. For instance, in a given course, there are often students from several majors: CS, Applied CS, EE, ME, or others outside engineering. Given that there is a spreadsheet column for "major" (or other subgroup indicator), the app notices these addition columns and makes it possible to selectively graph data from a subgroup. For instance, if the "Major" column exists, graphing

controls would default to “all students in course”, but there would be a pull-down menu with the label “Graph only those with Major=” and a way to select which ones you want included.
- Ability to configure a list of “ABET monitors”; these are people whose job it is to make sure ABET is getting done. With this list of names/emails, the system monitors ABET data in the archive. It sends a notification email the Monitors for key events, e.g., when ABET data for a course has not been filled in by X weeks after course end, or when performance indicators for a course go below some specified minimum.

This is just a first envisioning of features to be provided. Additional features and requirements should be captured from the clients, as well as based on the team’s evolving understanding of end user needs as the project progresses.

Knowledge, skills, and expertise required for this project:

- Familiarity with at least one programming language with data visualization capabilities (e.g. current code for an existing tool is in python, R would also be acceptable, or Javascript if a web-app approach is taken that relies on it)
- Familiarity with Excel and common data formats (csv, xlsx, json, etc.)

Equipment Requirements:

Free/open-source software tools should be preferred, for easy installation on any computer.

Software and other Deliverables:

- A fully-functioning application, as outlined above, installed and tested on a platform of the client’s choice.
- A “system administrators” manual that details step-by-step how the system can be installed on a platform of the client’s choice, as well as how to perform basic configuration and maintenance.
- As-built report, that carefully documents requirements, design decisions, and implementation details. Should allow future team to easily pick up where left off.
- Complete professionally-documented codebase, delivered as a repository on GitHub. There should be special attention to the API usage documentation, which will be used by third-party.