Collecting Student Data for Accreditation Assessment

Michael Ringenbach

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Masters of Science

in

Computer Science

APPROVED:

Stephen H. Edwards, chairman

Manuel A. Pérez-Quiñones

Calvin J. Ribbens

Keywords: Assessment, Accreditation

Abstract

Collecting Student Data for Accreditation Assessment

This paper seeks to identify one of the key problems faced by academic institutions seeking accreditation. The accreditation process requires academic institutions to conduct a self-study analyzing how well a given program is meeting the learning outcomes the accreditation board uses in its assessment. This self-study by schools often contains qualitative or subjective data and does not directly correlate the learning outcomes being measured to student performance. The lack of quantitative measurements at a granular level means that it is difficult for the academic institution to prove that it was effective in meeting a particular outcome.

I propose in this paper a tool that is both efficient and effective in capturing quantitative data at the student level. The tool maps specific coursework to learning outcomes and shows how students performed towards that outcome over the duration of a particular course or program. Additionally, the data collected by the tool can be used to assess course and program design.

Table of Contents

Contents

List of Figures

Chapter 1: Problem Statement and Previous Work

1.1 Background

The process of accreditation by outside organizations is one of the primary means for measuring the quality of a post-secondary educational institution's curriculum. These accrediting organizations develop a number of learning outcomes that they ask educational institutions to meet in order to receive the status of being accredited. The organization responsible for accrediting engineering and technology programs, ABET Inc., defines accreditation as "a non-governmental, peer-review process that assures the quality of the post-secondary education students receive. Educational institutions or programs volunteer to undergo this review periodically to determine if certain criteria are being met." [1]. The process of accrediting at the program level is referred to as specialized accreditation.

Accreditation is of vital importance to everyone involved. Institutions of higher education value accreditation because - as the only official indicator of its commitment to quality - it is often tied to grants and funding opportunities. Students as well as potential employers value accreditation as it is perceived as an indicator of the education a student receives. It is an indication of the weight that many employers are placing on the accredited status that many hiring opportunities require a degree from an accredited program and/or university.

Coupled tightly with the word "accreditation" is the word "assessment". Assessment is the means by which an educational institution measures the quality of its programs and thus gains an accredited status. Assessment is crucial to the faculty and administration; it is the sole tool at their disposal for showing to accreditation boards that their program is of sufficient quality to warrant the accreditation stamp of approval. Even non-accredited universities seem to be using accreditation standards to internally measure the quality of their programs [13], indicating the importance that most educational institutions put on the effort put forth by the accrediting organization in designing the accreditation standards.

The major difficulty with assessment is highlighted by the difference between what is being taught by faculty and what is being learned by students. Administrators can easily map accreditation outcomes to the individual courses within a program and thus meet all of the requirements given by the accrediting organization. However, that style of assessment tells accreditors nothing about whether students are actually learning or improving their skills through this program. Similarly, looking at the numbers of graduates of a program gives no indication that any given student has a comprehensive understanding of all of the accreditation outcomes. To achieve accurate assessment, performance data for students must be recorded, measured and analyzed.

The recording of performance data can come in many forms. Students and professors alike have often been asked to give reflections on the quality of their education, either at the course level or at the program level. Comprehensive and standardized exams have been offered to assess program quality. However, both of these approaches have significant problems.

1.2 Definitions

Before continuing the discussion, certain key terms need to be defined. The definitions used are quoted directly from ABET's site[28].

- **Accreditation**: a non-governmental, peer-review process that assures the quality of the postsecondary education students receive.
- **Institutional Accreditation**: an accrediting process that evaluates overall institutional quality. One form of institutional accreditation is regional accreditation of colleges and universities.
- **Specialized Accreditation**: an accrediting process that examines specific programs of study, rather than an institution as a whole. This type of accreditation is granted to specific programs at specific levels. Architecture, nursing, law, medicine, and engineering programs are often evaluated through specialized accreditation.
- **Assessment**: processes that identify, collect, analyze, and report data that can be used to evaluate achievement.
- **Objectives**: statements that describe the expected accomplishments of graduates during the first few years after graduation.
- **Outcomes**: statements that describe what students are expected to know and be able to do by the time of graduation.
- **Evaluation**: process of reviewing the results of data collection and analysis and making a determination of the value of findings and action(s) to be taken.
- **Outcome Pair:** a mapping of a specific program outcome to a specific and correlated accreditation outcome. Specific to the Computer Science department at Virginia Tech.

Throughout this thesis, I will be using the term accreditation to mean specialized accreditation as conducted by the Accreditation Board for Engineering and Technology, Inc. (ABET). While the results of this research are targeted for ABET accreditation and Computer Science programs, other programs and accrediting organizations may be able to benefit from the results as well.

1.3 The Accreditation Process

Before discussing the major problem addressed by this research, it is beneficial to describe the process an educational institution will undergo when ABET is evaluating their accreditation status. The process involves several steps beginning with a complete self-study performed by the educational institution. Upon completion, this self-study is reviewed by a panel of ABET evaluators. This panel of evaluators then visits the campus of the university and engages faculty, administrators and students in a fact-finding mission to determine the overall quality of the institution's educational program. In addition to discussions with the various parties involved, the evaluating panel also reviews documents that support the results of the self-study [27].

1.4 Problem Description

Abunawass [27] focuses on the need for institutions to complete and review the self-study to ensure the accuracy of the data presented, emphasizing that the department be able to provide any and all supporting documentation that is requested. However, much of this self-study is overhead for the educational institution [18]; the effort needed to perform an accurate and conclusive self-study generally has no immediate benefit to the faculty with regards to how they teach their courses and interact with students. Another problem with a self-study is that the data it provides is often only quantitative in nature and does not directly correlate to the learning outcomes that ABET uses to assess an educational program. Examples of this are using letter grades in a course or counting the number of students who graduated with a given GPA. These measures, while useful for a birds-eye view, don't necessarily give an indication of whether a student has a sufficient mastery of any given ABET outcome.

A prominent difficulty with assessment is the inability to measure the quality of a program in terms of student learning and improvement. First and foremost, the problem of time management is critical. Professors, students, administrators - everyone already has many demands on their time and despite the importance of accreditation, no one seems to have the time to fill out cumbersome surveys or take exhaustive exams. Some institutions have reported that they felt asking students to take exams for measuring student learning was "piling on" extra work for them[22]. They felt the students would not take the exam seriously. The same study also found that educational institutions felt external exams may not be fair in that they were being compared to many other institutions of a differing caliber.

Similarly, it found that various survey types were viewed as a burden, some types of surveys having an "abysmal" return rate, offering little usable data.

Other problems exist as well. The student performance data recorded is often mapped to accreditation learning outcomes through an educational institution's course outcomes. The assumption is made that if a student has passed a course, they then meet the accreditation outcomes associated with the course. This approach leads to problems due to abstracting measurements away from the students. For any given measurement, there is no direct tie to whether a student learned enough material to sufficiently cover a learning outcome.

Out of the myriad of problems that exist within the scope of accreditation assessment, I have chosen to address the lack of having an effective tool for efficiently collecting student course work results and the subsequent mapping of student work to accreditation and program outcomes. Addressing this problem gives university administrators concrete, quantitative figures to analyze during the self-assessment portion of the accreditation review. Additionally, the collected data makes available a wide variety of research opportunities.

1.5 Introduction to Solution

I propose a system that attempts to solve many of these problems. The solution proposed is to have professors store within their gradebooks the mapping between student performance and learning outcomes. I will then offer a tool by which professors or administrators can upload gradebooks and immediately receive assessment feedback and analysis. By mapping the accreditation learning outcomes to the grades in a professor's gradebook, measurements are taken over the length of a given course directly at the student level. By offering affordances and options in how to record the mappings, the burden imposed on any given professor's workload should be negligible. The aim is to make the alterations to a professor's grade-keeping methods minimal, thus making the solution a minimally invasive one with a higher payoff relative to other proposed solutions.

Chapter 2: Literature Review

Outcomes-based assessment has been an ongoing topic of research for many years. Here I will look at a number of studies and projects that have tried to address the problems described. I will then go on to show how the problems are either not satisfactorily addressed or solved in a different manner.

Numerous researchers in computer science education have addressed the issue of assessment [2, 4, 5, 9, 12, 14, 18, 20, 22]; some to explain the process[16], others to devise certain methods[2, 6] and a few researchers introduced tools they claim would simplify or reduce the burdensome task of assessment[3, 7, 24, 25]. Within the presented tools, a large number of papers use the term assessment interchangeably with grading; hence they use automatic assessment as meaning automatic grading [4, 21]. While this is a correct use of the term, it is significantly different from what we mean when we use the term assessment. These papers present methods and tools that are not strictly tied to accreditation assessment and learning outcomes; yet the pitfalls, lessons learned and even the tools described in papers on this topic are often applicable or adaptable to accreditation as well. It also allows for us to see how research in a related field has developed and continues to develop. Focusing on the type of assessment that is related to program development and accreditation, there exists a category of tools that more closely resemble Course Management Systems than assessment tools [15], with the focus being on course material distribution and student information presentation as opposed to the actual automated collection of fine-grained student information (i.e., the actual grades, student information). I will examine several of these tools and use their findings to examine our claim that a tool to support accreditation assessment must be tied to course management tools to obtain ultimate efficiency in reduction of workflow redundancies.

Blaha and Murphy [5] raise several key issues regarding the different aspects of assessment. When talking about assessment, it is important to know what exactly is being evaluated. The authors list several possible targets of assessment such as students, faculty and curricula. While several of the targets can overlap and can be measured using similar techniques and data, it is still important to distinguish what the purpose of an assessment is and what the goals of each target are (since the goals of assessment can be different for differing targets). In addition to the paper suggesting that institutions know how and when assessment will be performed, they also highlight the importance of knowing how the evaluation data will be used to alter the target of the assessment. Throughout my paper, I will be focusing on accreditation assessment, which is typically an assessment of the curriculum as a whole. Many of the papers I reviewed explicitly mention that faculty and students often felt as though they were the subjects of the assessment as opposed to being part of the assessment process. While I believe that the system presented in this paper can avoid making faculty feel as though they are the target of the accreditation assessment, a full study has not been done to confirm this. I am cognizant of this potential hazard and will expound on it more in the future work section.

Blandford and Hwang [6] define what they describe as five easy but effective assessment methods. The five methods are described as easy (which the authors use to mean non-burdensome to the faculty and administration) and effective, meaning that the methods will result in the improvement of a course or the curriculum as a whole. The five methods described are 1) The concept of Assessment Day (setting aside 4 hours each term for the faculty to discuss the semester in terms of course assessment); 2) Personal class assessment, wherein each faculty member completes a web survey for each class they taught; 3) Sampling of student work, which involved using one of two methods to get a feel for how students performed in a class; 4) Faculty interview groups, allowing faculty to conduct a series of interviews with a sampling of students allowing for direct feedback from students; 5) Student focus groups in which a leader will begin with a set of questions and then adapt the questions to fit the discussion and to allow for free-form response. The problem with these techniques is that they are all qualitative in nature, save for the student work sampling. For the other four methods, there is no direct means of measuring progress across terms or comparing one term to another, or even one class to another, in an objective manner. And while the paper describes these methods as easy, no real description or implementation was offered for how to collect, record and measure student data. The solution presented in the paper was to have faculty determine which student work "best illustrates that students are meeting a particular outcome." This method seems to highlight one of the problems mentioned throughout this paper; namely that instructors who "teach to a standardized test" are collecting skewed data not necessarily representative of true performance. In the case described by Blandford, the faculty begin with the assumption that students are meeting an outcome and that they must simply show the accrediting organization that they are meeting it, as opposed to finding a method which will give an accurate representation as to what percentage of students are demonstrating various levels of mastery for a given outcome. This premise of "teaching to the test" will be one addressed by our project.

Pardo [20] in 2002 was one of the first to try and bring computer aided assessment to the web. In his paper, he describes an architecture for allowing numerous user types to interact with a single system that would facilitate the collecting and grading of assignments. Professors, students and graders would all have access to a web-based interface in which they could each perform their respective tasks. Students would be able to complete assignment submissions. Graders would be able to use a variety of grading methods to assess the students' work. Professors would be able to administer the assignments and view the assessment. This paper raises several of the key issues (such as the need to support multiple assignment types) that we are still trying to address in accreditation assessment.

Gowan (et al.) [12] created A Configurable Assessment Information System (CAIS) that works to bridge the gap between program outcomes, skills and the courses and students. The system primarily focuses on the flexibility of program design, accommodating both course-based programs and problem-based programs. The assertion is made that courses are designed to teach certain skills and that those skills map to program outcomes. Transitively, the program outcomes are able to map directly to the courses, though their system allows for a direct link to courses in addition to this transitive mapping. The method of recording measurements relies on user configuration of the system, allowing the user to indicate what type of measurements they will be recording (e.g. survey-based or exam-based) and then an appropriate table is dynamically generated and added to the system. Very little information is given in the paper about how the actual measurements are entered into the system by the user. While this system does move towards the idea of tracking measurements of student learning, the flexibility of its data model creates a high level of overhead for the administrators that will have to use the system. In addition, two other major differences can be found between our system and the CAIS. First, the student entity is optional in their data model, allowing for a loss of granularity in recording and tracking individual learning. Second, the types of measurements gathered can still include surveys and other qualitative assessments. This can facilitate the "folk" pedagogy that persists in assessment as opposed to moving towards a more quantitative means of assessment.

Aasheim, Gowan and Reichgelt [2] provide another process for accreditation assessment that is comprised of four separate instruments for assessing. Three of these - graduate surveys, employer surveys, and student exit surveys - all focus on qualitative, indirect measures of program assessment. They compliment these three processes with a quantitative measure in the form of a table that maps course outcomes to assessments. Within this table, each course outcome is provided and the course instructor is expected to provide the rest of the information. This information includes how each outcome is covered in the course, what measure will be used to assess it and then what percentage of students passed (received a grade of 70% or higher) that metric. While a tool could easily be written to help alleviate the burden this places on professors, this solution falls short on several fronts. First, no data is being recorded at the level of individual students. The only assessment data that is being quantitatively collected is at the course level. This creates a loss of data for how individual students are performing and data mining cannot be performed to find how individual students have changed or improved over a course or program. While the other three instruments for recording data do occur at the student level, this data is all qualitative in nature and does not provide a concrete metric for assessment.

Holden [16] uses a very similar methodology for accreditation assessment, choosing to map student performance directly to the desired outcomes. In his study, he uses an introductory database course wherein several outcomes are measured using the final exam questions that pertain to those outcomes. He compares these measurements for two specific years. Holden specifically mentions that his process provides a minimal burden to the instructor since no additional requirements are made on his teaching schedule. However, no tool is developed or provided for gathering this data, relying instead on a manual aggregation process. Additionally, this is another process which does not make plans for accommodating individual student record-keeping, choosing instead to record measures as a percent of the class that passed the assessment metric.

Longenecker and Feinstein [14] describe several techniques to be administered by a national vendor-neutral assessment organization. The Institute for Certification of Computing Personnel has a division called the Center for Computing Education Research (CCER) which administers an academic assessment exam. This exam requires academic institutions to create local objectives, mapping individual components of the curriculum to exam questions. This would allow universities to use the exam for internal course and curriculum refinement while simultaneously allowing the CCER to handle the assessing of accreditation outcomes. The burden of accreditation is thus completely removed from the hands of the academic institutions. This approach has all of the potential problems inherent with a standardized exam that are currently found in primary education, such as the loss of instruction for non-core or non-measured skills and the inability to account for regional differences in programs.

Another process for accreditation assessment is described by Maxim [19]. It begins by mapping the ABET accreditations to each of the degree programs offered at the University of Michigan-Dearborn where the study was conducted. Each course was then mapped to one of the accreditation outcomes as well. Students entering a course must take a prerequisite exam prepared by the professor for that course. This prerequisite exam is used for two purposes: to ensure that all students meet the minimum qualifications for taking a course; and by an assessment committee to ensure that courses are properly meeting the objectives. Additionally, student and employer surveys are conducted to help ensure that assessment goals are being met. The assessment committee meets at the beginning of each semester and reviews all of the available data and makes changes as appropriate to the course curriculum. While this process covers many of the problem points with assessment, such as a lack of quantitative measuring of student learning, it does create a significant burden on professors, as the paper notes: discussion among the faculty regarding parts of the process have been "lively" though generally considered a worthwhile time investment. Further work on this project could automate a significant part of the process alleviating that burden but would still fall short in several areas, namely that individual student data would still not be recorded (though alterations to the process could add this). Additionally, there is the danger that by adding an extra measurement to each course, professors run the risk of "teaching to the test" in an effort to inflate the performance of their students.

Yue [26] describes a course-based assessment model wherein ABET outcomes are indirectly mapped to courses through the course objectives. To assess each course, a suite of approaches is then used, including a course exit survey, an evaluation of the capstone project, and a per course examination evaluation. In this final tool, the examination evaluation, professors would identify from the course a suite of questions that adequately covered the course objectives, and then analyze the scores of the students, thus providing fine-grained quantitative student data. However, no actual tool was offered to the professors to assist in this and thus the paper records this model as not being used very often and are reconsidering its use in future assessments. This also was the only quantitative model being used for assessment in the paper, the other two both being qualitative in nature.

Yang and Beaubouef [25] provide an interesting tool that ties directly into the Blackboard Course Management system. Their tool will use course documents to dynamically generate a survey for course assessment. This survey is then administered to students via Blackboard sometime during the course. The inclusion of their tool in to a course management system provides a large number of benefits that are mentioned in the paper, such as the minimal amount of extra work that would be incurred on professors, and several that are not mentioned in this paper, such as being able to track student responses over time and comparing sections of the course to one another. However, this tool is purely a qualitative indirect measurement of student learning.

COMPASS [3] is another tool that recognizes the value of tying objectives directly to student work. The COMPASS system is a module for Moodle, a web-based open-source course management system. The module allows for professors to create assignments and then tie those assignments to learning objectives associated with the course. Each of these assignment-learning-objective mappings is then weighted to indicate at what competency level a student should be performing. Professors are then given a series of input options to describe and categorize the assignment. While the COMPASS system provides many benefits to assessment, it still fails to record student learning at a sufficiently granular level. By only allowing the objective mapping to occur at the coursework level, a certain amount of detail is lost. A student that gets eighty percent of a coursework correct could have missed every question relating to a particular objective but the system would not be able to account for that, weakening the score the student should have received for other objectives and strengthening the assessment of the problem objective. It is this loss of granular data recording that I am striving to correct.

Booth [7] in his 2006 paper creates a very detailed data model which he proposes as the basis for future assessment models. The model tracks each offering of a course, and each assignment within the course. Student answers to the assignments are stored along with the assessment meta-data. The assessment meta-data is mapped to course objectives, and the course objectives are in turn mapped to ABET-specified program outcomes. While this is a fairly comprehensive data model, it does not offer a practical approach for how to implement a system to utilize this data model. Also of note is that the data model has several abstractions that could prove cumbersome in an application, such as the course objectives not being tied to a course except through the program outcomes. The assessment meta-data entity is only a summary of the assessment measures and does not directly map to program outcomes, choosing instead to form that relationship through the course outcomes table. Problems could arise should any of these relationships not map directly one-to-one. Another problem could be the ambiguity of the assessment entity and the assignment entity. This forces a certain limit as to how refined measurements can be, preventing individual questions from being mapped directly to the program or course outcomes. Given that most assignments typically apply to more than one outcome, this could be a severe limitation.

Booth (et al) later improved upon the design described above developing a tool called WebSubmit [8]. Many of the problems mentioned above are resolved in an updated version of the data model, though the ABET outcomes are still mapped to individual courses through the program or course outcomes. This indirect mapping of ABET outcomes to the student assignments could lead to a loss of specificity as instructors create mappings that are a "best fit", potentially skewing the resulting measurements. The interesting part of this tool is the method it uses to actually assess the student submission. When a professor creates an assignment, he also creates a grading rubric for it where each item on the rubric corresponds directly to a course outcome. When an assignment is electronically submitted by a student via the web site, the professor can view the submission and then check each item on the rubric that the student correctly addressed. The grade for the assignment is then determined by the number of items checked. This ties every element of every assignment directly to the assessment of the course and thus also to the ABET assessment. This however, causes several problems. It embraces the idea of "teaching to the objectives". Every question must map to a course objective, thus forcing professors to ask questions in such a way as to make sure that they will accurately map to an objective. This is not always the case, as many questions a professor could ask will not be so clearly defined in terms of course objectives. Another significant problem is the means of submitting data. By only assessing assignments electronically submitted by the student, only a portion of assignments will count towards the assessment (unless all exams, homework, projects, and in-class assignments are submitted electronically). This presents another problem of skewing data. If a student tests well but doesn't do well on homework, yet the final exam is proctored in a standard classroom environment, the WebSubmit tool doesn't allow for the exam to count towards assessment. This could then inaccurately record how well a student has mastered a given objective. If a student performs poorly on several earlier homework assignments but then does very well on an exam not submitted electronically, the WebSubmit tool will show the student as not meeting that objective, despite the potential that he did well on the exam. Another problem with this tool is that it creates a significant amount of work for the professors. The system seems inflexible in that all coursework must have a rubric and that each element on the rubric must map to an objective. Performing these tasks could take a large amount of time, thus reducing the likelihood that professors will take the required amount of time to do the task correctly. This is one of the main problems that I attempt to overcome in our system: the need for a timely and efficient means of interacting with the assessment system. The final problem I will mention is that the grading tool does not allow for partial credit. Oftentimes when a question is worth multiple points, professors will give partial credit to signify that the student understood part of the concept or was able to correctly address some portion of the problem. This "all-or-nothing" means of grading questions does not accurately reflect a student's mastery of a given objective - developing, mature or proficient - and could result in an inaccurate assessment. This is another problem that I am attempting to solve with this system by recording both the maximum value that a question is worth and what percentage of points the student earned on that question.

One of the most advanced student-based outcomes assessment methods available is one proposed by Winters [24]. He and Payne used several data collection tools to collect student data from a variety of courses. For each course, a list of 10 to 12 outcomes were developed. These outcomes were then mapped to the course questions in a relevancy matrix R. To ease the burden of filling in all of the numbers in what could be a fairly large matrix, the authors used several data mining and analysis techniques to fully populate a partial matrix filled out by the professors. The paper indicates that they would ask professors to fill in 10-20% of the relevancy matrix by which they could then fill in the remainder with relatively high accuracy. This matrix was then compared against a partially-filled student answer matrix (where each row was a student and each column represented a coursework). Cells in the student matrix that were empty would indicate that a student failed to complete an item or receive a grade for it. To completely fill in this student answer matrix, they generated a predictive model allowing them to reasonably estimate what grades students would have received had the assignment been completed and graded. In the area of data collection, the paper describes two tools, Agar and MarkSense, that were developed specifically to grade and process the student assignments. These tools can be both complementary to existing grading styles or be used as full replacements for manual grading. The tools focus primarily on the automation of repetitive grading tasks for assignments and have been quite successful in the courses in which they have been used. The one major drawback to the system is that they do not address what happens for assignments in which their tools are not used. Rather than looking at a professor's gradebook to process the grades, it seems that they are instead relying on the grading tools to be used to collect the raw scores, which is then outputted in a pre-determined format. The inflexibility of the system is a potential barrier to adoption within the department. This particular barrier is one that has been identified in the past and is one that we are keenly aware of and are trying to address.

Booth [8] acknowledges this importance; however he uses a larger granularity and does not include the student entity in his database which significantly hinders the ability of an assessment to retrieve, compare or follow data across different offerings and throughout student variations.

Yang and Beaubouef [25] propose a tool for generating web surveys for assessment purposes. This tool is designed for students to evaluate the class they took for assessment record keeping and improvement. While apparently different from what I am proposing, it emphasizes “saving faculty valuable time.” The major difference between Yang and Beaubouef's system and mine is the type of data gathered. While qualitative data can definitely be useful, it does not offer the precision and concrete evidence of student improvement that quantitative data offers.

All of the aforementioned papers are certainty valuable contributions as assessment tools. However, I believe they do not directly address the overhead in time and work effort which the manual gathering of a tremendous amount of student data creates. The Web-CAT assessment tool is specifically designed to address this deficiency.

Chapter 3: Web-CAT (Course Assessment Tool)

With the main goal of capturing detailed student information in mind, we are proposing that professor's record the mapping of Program and Accreditation Outcomes to assignments and questions in their gradebooks. We will then offer a web site to allow professors to upload their gradebooks in Excel format and store those grades. Reports can then be viewed either by the professor or administrators breaking down how well the outcomes are being met at the student level, course level, program level, and several other options.

3.1 Solution Overview

I am proposing a solution that will address the problems mentioned above in the following ways:

1. The recording of quantitative student performance data will provide a measurable figure to show how much a student or group of students has improved over the course of earning a degree or completing a program.
2. By taking granular measurements throughout a course and taking the measurements as part of the course materials, the measurements will more accurately reflect student learning than either standardized exams or traditional end-of-semester course assessment surveys.
3. The flexible and integrated method of recording outcome measurements in the gradebook will be acceptable to most professors and have a minimal impact on their traditional bookkeeping methods.

To support these assertions, I have enhanced the Assessment module already found in the Web-CAT automatic grading and submission tool currently used within the Computer Science department at Virginia Tech. This module will allow professors, their TAs, or administrators to load the gradebook for a course into the system via a simple online form. I will examine the process by stepping through the typical workflow of the system.

3.2 Solution Implementation

When implementing the solution, I was presented with the opportunity to expand an existing tool known as Web-CAT, an automatic grading system for programming assignments. Web-CAT has been a tool used successfully in the Computer Science department at Virginia Tech for several years. Expanding Web-CAT proved very attractive for several reasons. First, the professors are already familiar with the Web-CAT system and interface. As an existing part of many professor's teaching toolkits, integrating the new tasks in the assessment module into a tool already found in the professor's workflow would greatly ease the time overhead generated by course assessment. In addition, the data model for Web-CAT is fairly well-developed and robust. Several core entities in the system could be used, eliminating a potential redundancy in data collection, as well as removing the need to develop redundant tools for collecting or assimilating this information.

An Assessment module was being planned for Web-CAT prior to this project. The module originally contained surveys and qualitative reflections to be filled out by professors. The portions of the Assessment module described in this paper are separate from and complimentary to the existing module components.

3.3 Workflow Priorities

My main priority when designing the workflow was to minimize the amount of additional work imposed on the professors using this system. Professors as a whole are over-burdened with many demands on their time. Placing an additional burden on them that gives no direct benefits to them would mean the system would be used less than optimally if at all. To accommodate this, I placed minimal intrusion as the top end-user priority. To also help ensure the system would see use, I am including reports that professors will find useful for course and self assessment, in addition to reports that administrators will find useful for accreditation assessment.

There are two main tasks that can be accomplished with this system: uploading a gradebook and viewing student data. I will address each in turn.

3.3.1 Viewing and Uploading Outcomes

Within the system, a professor or administrator is able to view certain learning outcomes. Administrators are given the ability to upload a simple CSV file containing all of the departmental or accreditation learning outcomes. Professors are then able to view these outcomes. For each course a professor teaches, he or she is given a list of the outcomes - both program and accreditation - that are associated with the course[1]. They are also able to view these outcomes via the Web-CAT site.

[1] The details of how these outcomes are created and associated with a course is not necessarily relevant for this paper. However, once data from a course has been collected and evaluated, these outcomes may be changed. While I am aware of this aspect of assessment, it is not my focus. Research potentials for this topic will be covered in the future work section.

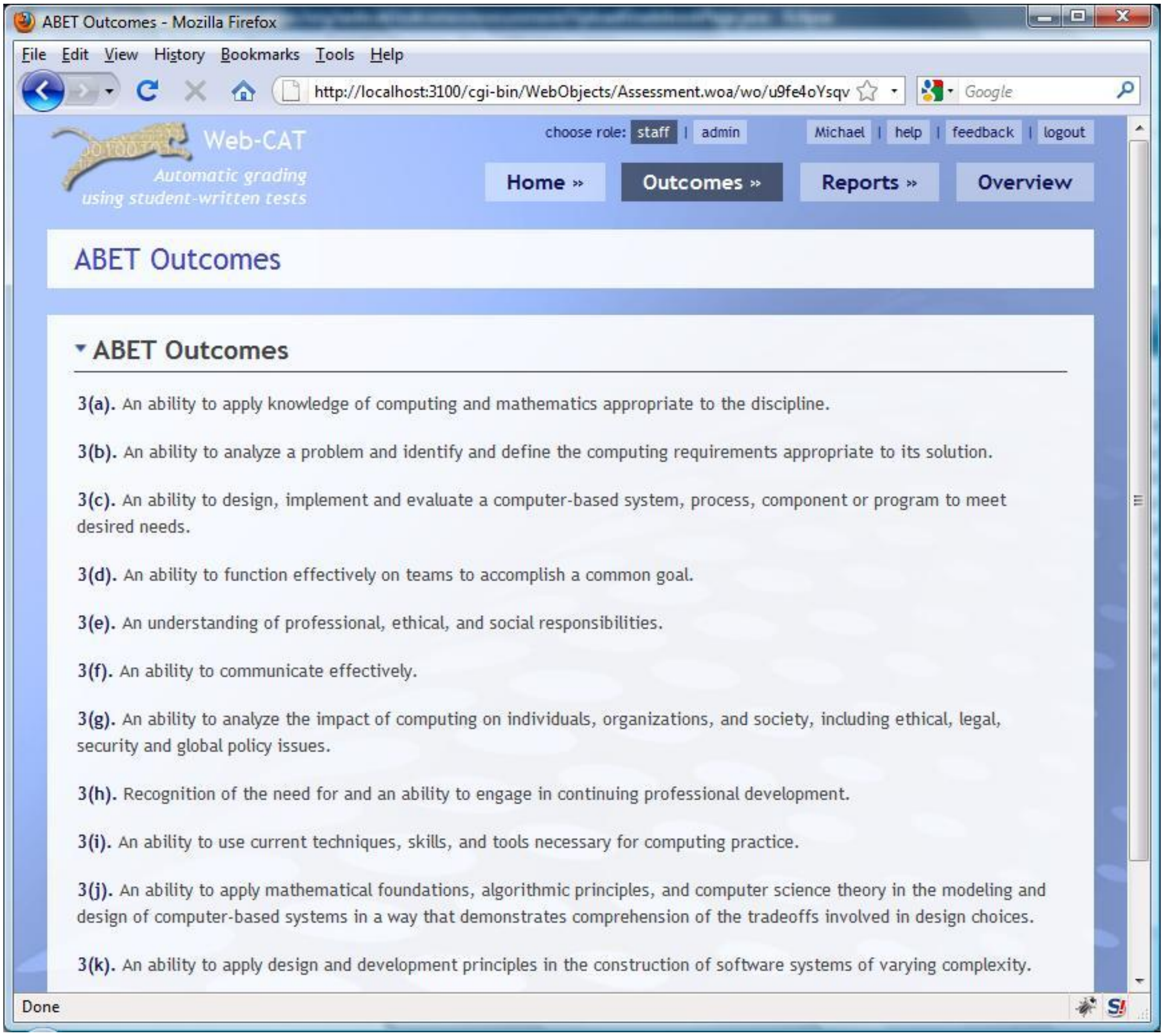

Figure 1: List of ABET Outcomes

These learning outcomes, as previously mentioned, are the primary metrics against which a program is measured. To properly and accurately analyze the quality of a program, it is essential to measure data that is as closely representative of the actual learning outcomes as possible. The departmental outcomes (fig. 2) are used primarily for internal assessment.

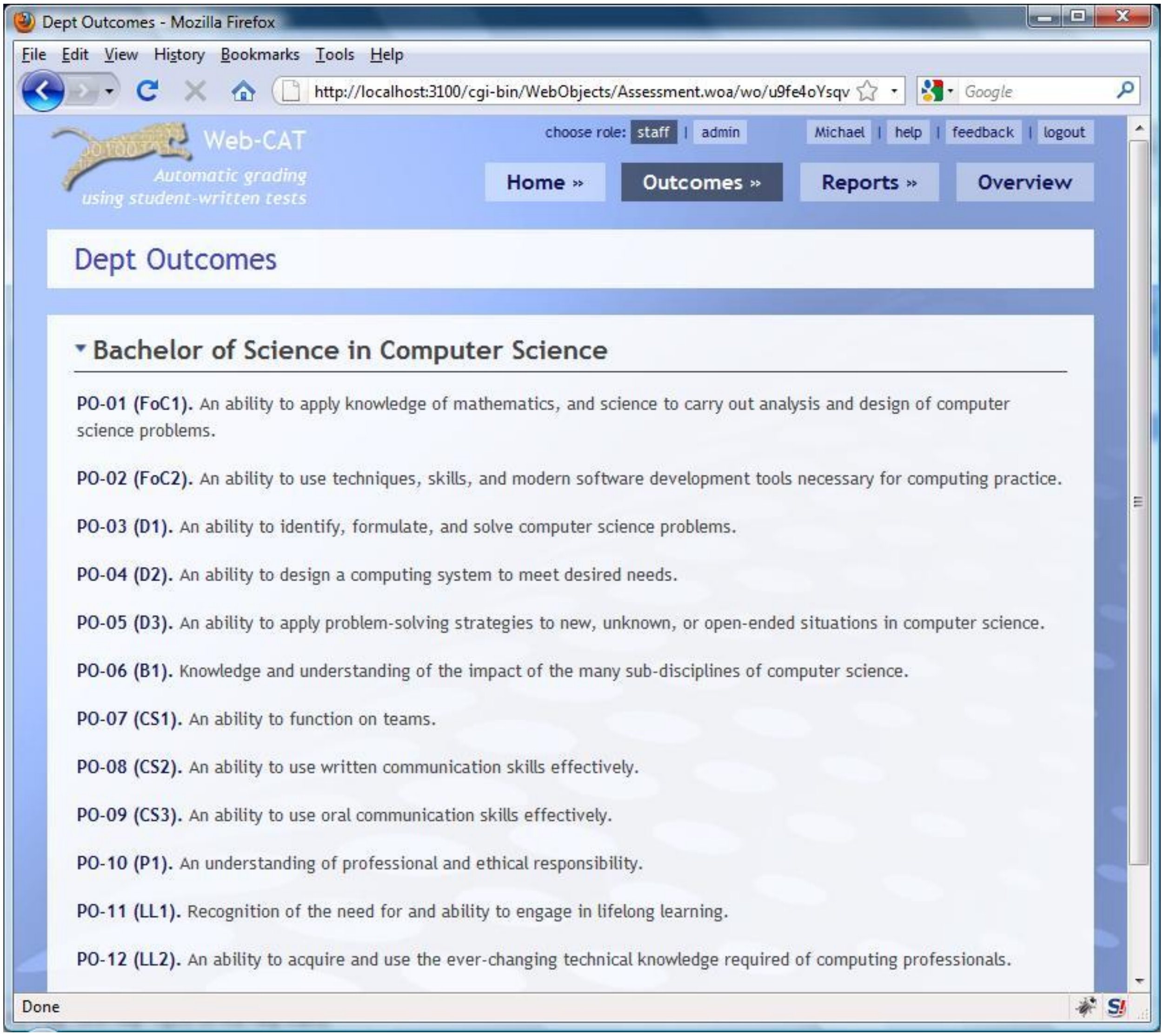

Figure 2: List of Program Outcomes

3.3.2 Creating a Gradebook in Excel

Certain requirements are imposed upon the user for how the gradebook must be kept in Excel. As of this writing, both .xls and xlsx files are supported. Internal to the Web-CAT system, I chose to parse Excel files using the Apache POI open-source library. It is fairly robust, easy to use, and is in constant development, which will hopefully allow for more document formats to be used in the future.

CS 101	GTA:	John Smith				
Student	CRN	HW 1	HW 2	Exam 1	Project 1	Final
Mapping		k4	c3	i3	4j	3j
Max Points		10	10	50	50	100
aaa@vt.edu	1	8	9	46	45	92
bbb@vt.edu	2	2	3	22	44	34
ccc@vt.edu	1	4	2	2	4	23

Figure 3: Sample Spreadsheet

The system expects students to be identified by either e-mail, PID, or 9 digit ID number. These three types of identifiers are the most commonly used identifiers in gradebooks. Identifying students based on name (last, first) was considered but discarded in the current incarnation of the project. Without other information, name by itself may not be enough to uniquely identify students in the system. Rather than risk data poisoning by misidentifying students, if name is provided in a column, the system will ignore it in favor of other identifying columns. Integration with a course management system would allow for tighter coupling of a course roster with a gradebook for that course.

Students must also be associated with a Course Number (CRN), which can be stored in a number of ways. Some professors may prefer to store multiple sections of a course in a single sheet. In this scenario, a column to the right of the student ID identifies the section number. If sections are stored on different sheets, the system will look for an identifier either as the sheet name or in a field at the top of the sheet. Additionally, the system will still look for an identifier column to the right of the student id if no other course identifier is present.

The max value for each assignment or question must be included in a row somewhere on each sheet. The data collected for each student must be stored relative to what the maximum value is. Without the maximum value for an assignment/question, there is no meaningful way of measuring the results.

An optional row for each assignment containing the Outcome Pair mappings may be included in the gradebook. This row, while optional, is the key function of our system. It is this row that will tie accreditation and program outcomes to individual student performance. After the gradebook has been uploaded to the system but before it is committed to the database, an opportunity to add these mappings will be presented to the user.

3.3.3 Uploading a Gradebook

The action of uploading a gradebook is not tied to an individual course, so no prerequisite actions must be taken beyond logging into the system. This also means that the action of uploading a gradebook can be delegated to another system user, either from the administrative staff or to a Teaching Assistant.

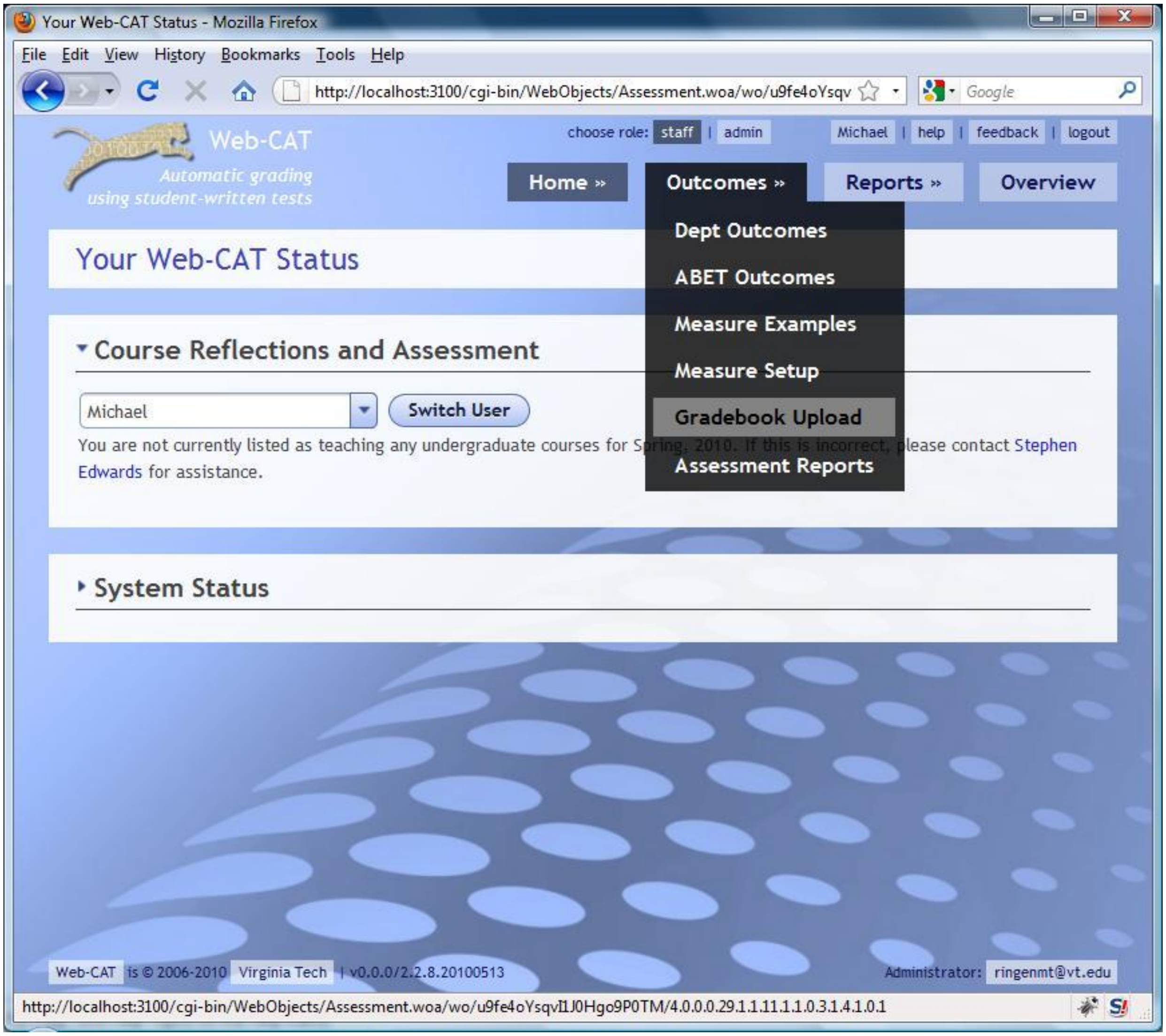

Figure 4: Web-CAT's main page with the gradebook option highlighted

From the main menu, the user will select the "Upload Gradebook" option. Selecting this option will take the user to a page with more detailed instructions for how to upload their gradebook.

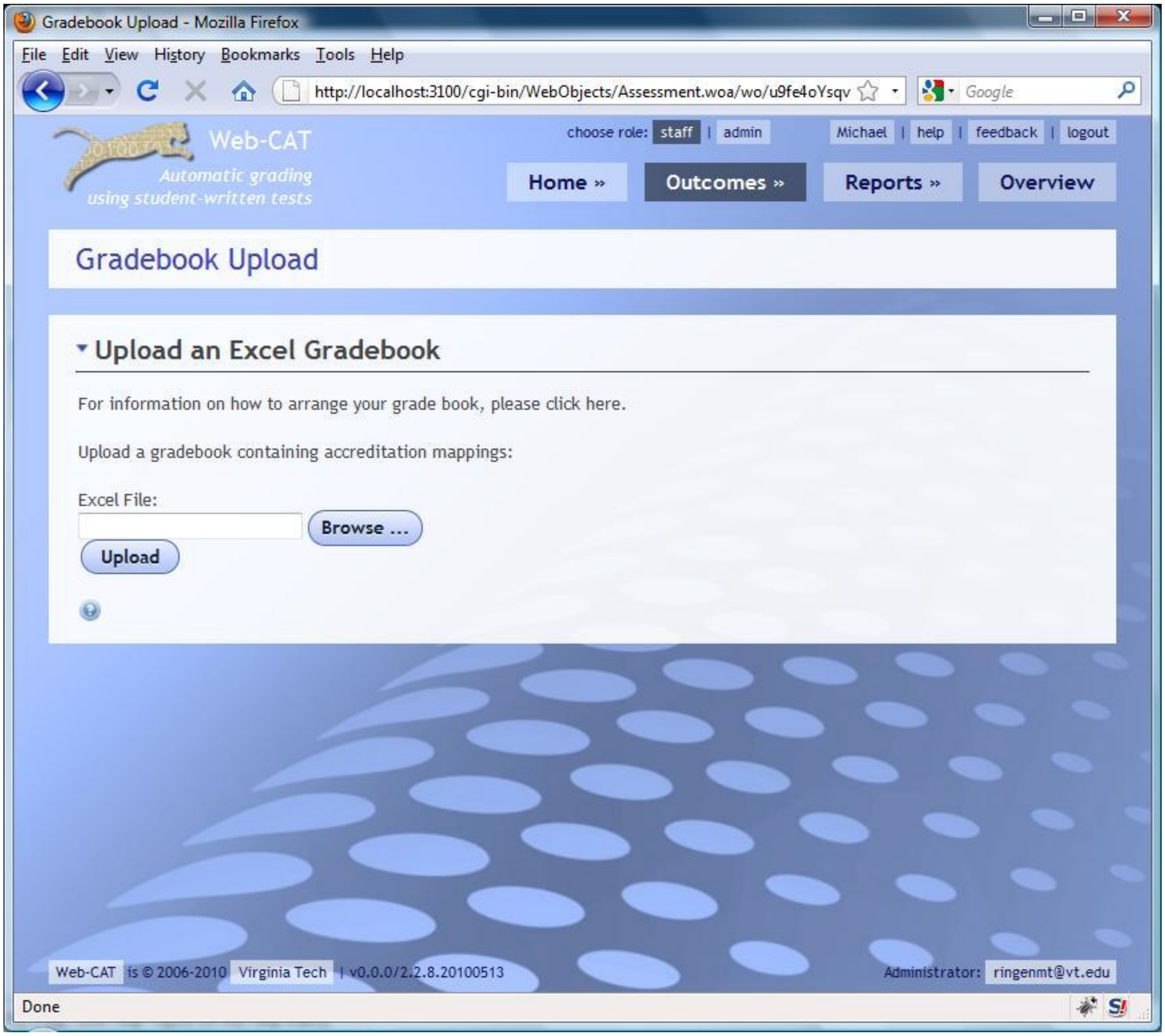

Figure 5: Upload Gradebook Page

Included on the page is a link explaining in more detail the options a professor has in the layout of his gradebook. The page allows the user to use the standard file browser to select his or her gradebook, stored as an Excel spreadsheet.

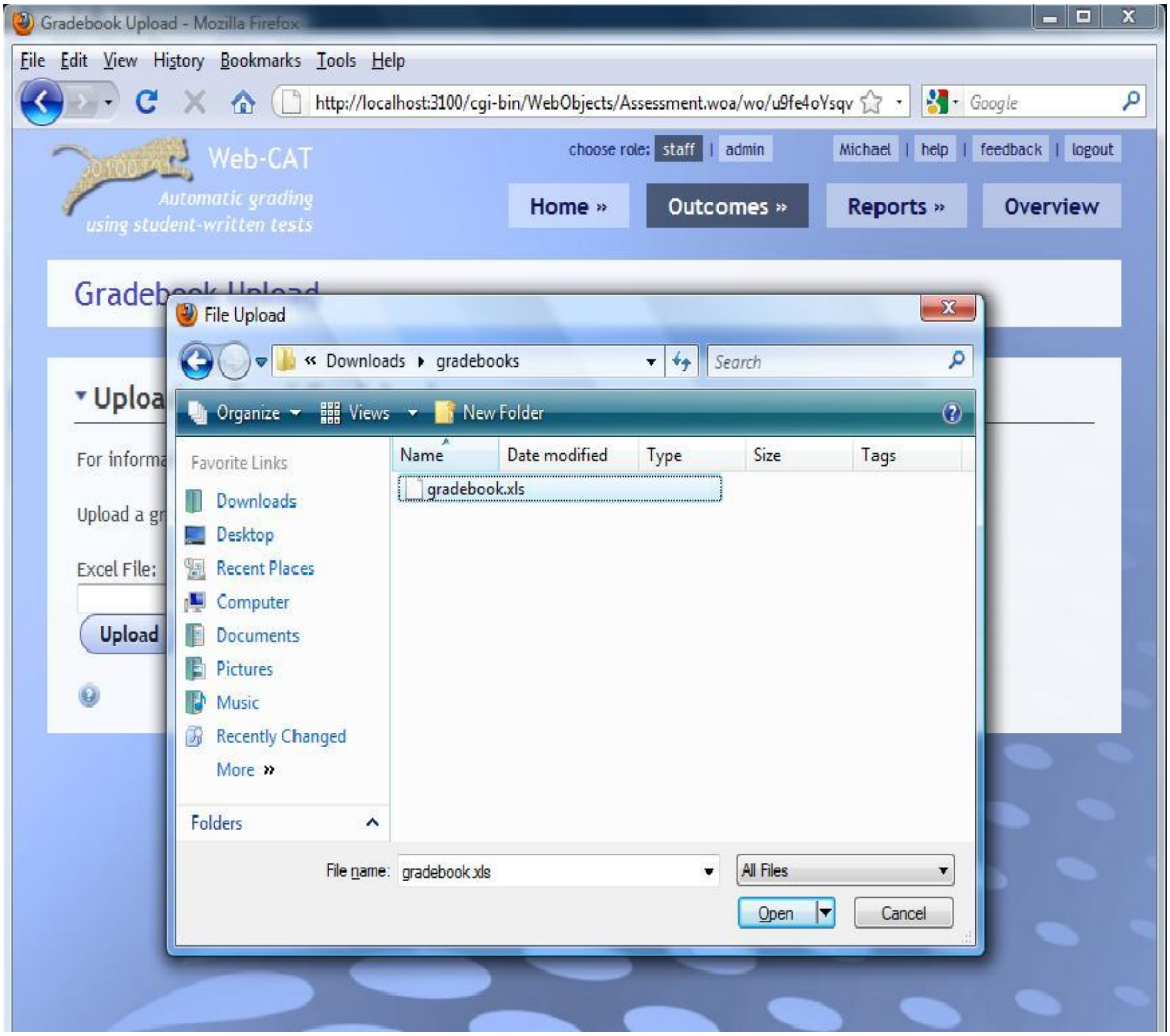

Figure 6: First gradebook upload page

Once the user selects a file and it has been uploaded, the user will be presented with all of the recognized grade sheets in the Excel file. For each sheet, the user will be able to add a letter-number Outcome Pair mapping to each assignment. The user is able to assign a mapping to as many or as few of the coursework columns as they wish. The more mappings that are added to the system, the more accurate the collected data will be.

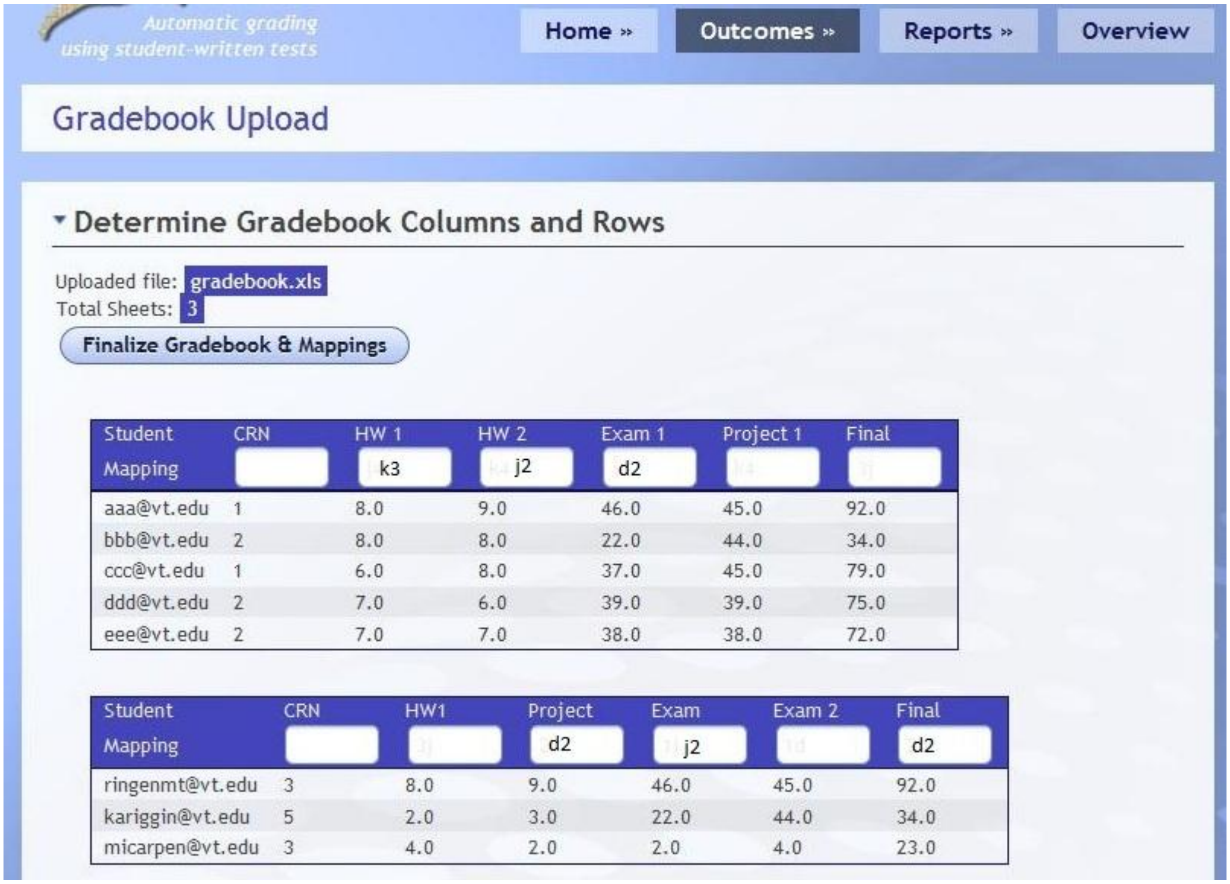

Figure 7: Gradebook upload and outcome mapping

When the user is finished editing the mappings, he will press the "Finalize Gradebook" button and the results will be added to the system. The user is then taken to a Reports page.

3.3.4 Viewing Reports

The second main task that the system provides is the ability to view analysis of the student data stored in the system. The user can access this task in one of two ways. If a user has just uploaded a gradebook, it will automatically take them to the reports page. Otherwise, a user will access the reports page via the option in the main menu.

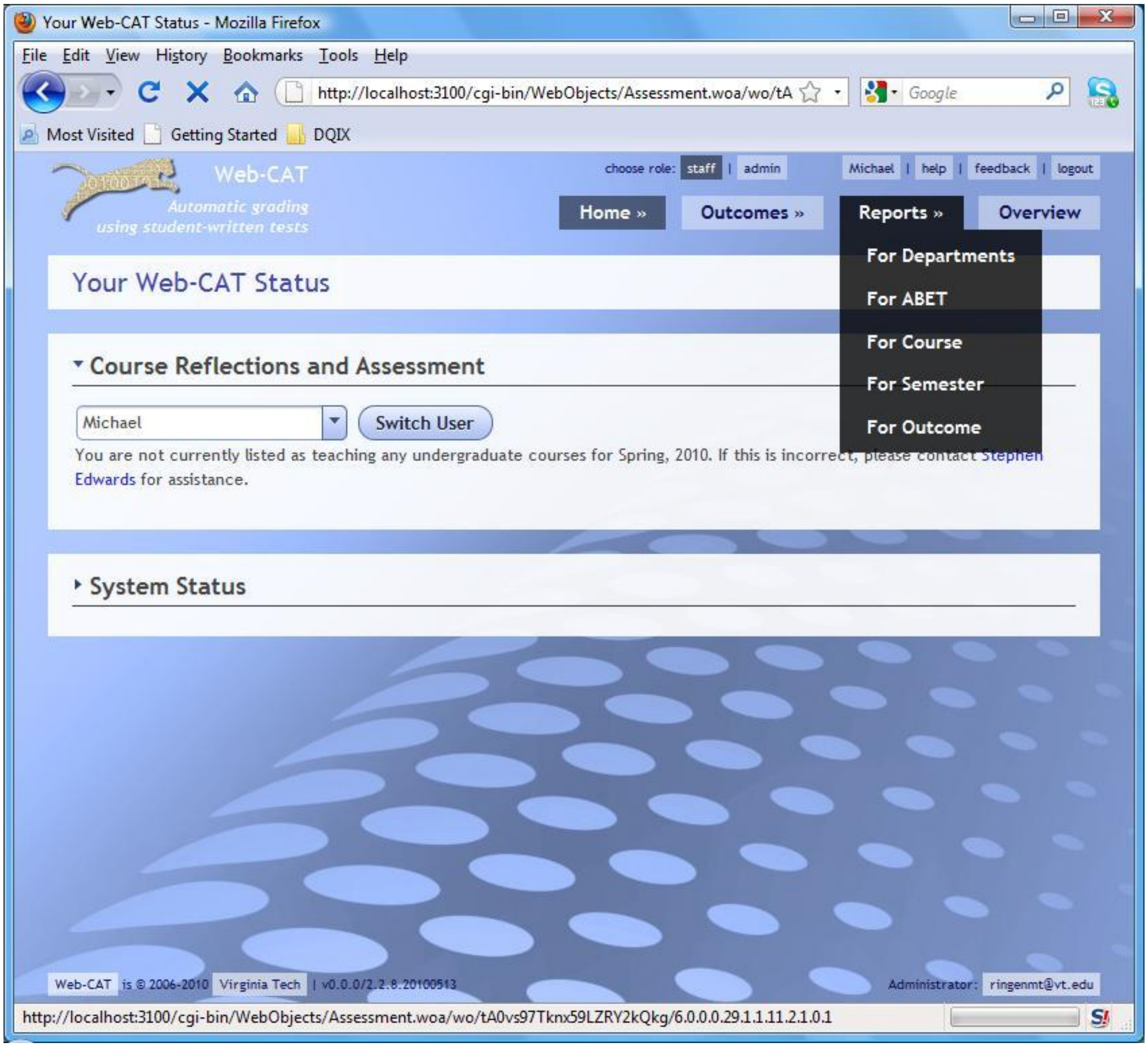

Figure 8: List of the Reports Available

The reports menu currently contains 5 options for how to retrieve data.

- For Departments: This report offers a breakdown of all outcome pairs currently being used by a given department.
- For ABET: This report offers a breakdown of all outcome pairs currently being used by the accreditation organization ABET. Future work will include support for other accreditation organizations.
- For Course: This report offers a breakdown of all outcome pairs related to a specific course. A menu option allows for the viewing of a different course.

- For Semester: This report offers a breakdown of all outcome pairs related to a specific semester, defaulting to the most recently completed semester. A menu option allows for the viewing of a different semester.
- For Outcome: This report offers a breakdown of all outcome pairs related to a specific program outcome. A menu option will allow for the viewing of a different program.

Each report has an identical layout for displaying data. The reports give a top-down view of all of the statistics for the various outcome pairs that have been recorded. It has two views, a summary view (shown by default) and a detailed view which will give a breakdown of each Outcome Pair, detailing how many measurements have been collected for that Outcome Pair and how many of those measurements fell into each of three categories: excellent, adequate and unsatisfactory, indicated on the graph by green, yellow and red color-coded bars. The excellent category indicates a high level of mastery for the outcome pair. The acceptable indicates a moderate proficiency while unsatisfactory is a sign that more work needs to be done in this category.

The three categories - excellent, adequate and unsatisfactory - have internally stored default cutoff points. The default cutoff for the excellent category is a score of 80% and the default cutoff for acceptable is 60%. In the future work section, we will detail a means by which professors can record what the cutoff for each category will be for a particular coursework. The system will then use these coursework-specific cutoffs over the default values when available.

The initial report in the list is the Departmental report (shown below). Each report can be viewed in one of two modes, Summary (the default) and Detailed. There are plans to add a third mode that offers a compromise between the level of data available in the other two modes.

The Summary view is the default view and will display a horizontal bar graph for the outcome pairs currently on record for that report. The graph will begin by indicating what the micro-label is for the outcome pair and then next to it display a bar broken down into color-coded segments giving a birds-eye view of what percentage of scores measured fell in to each range.

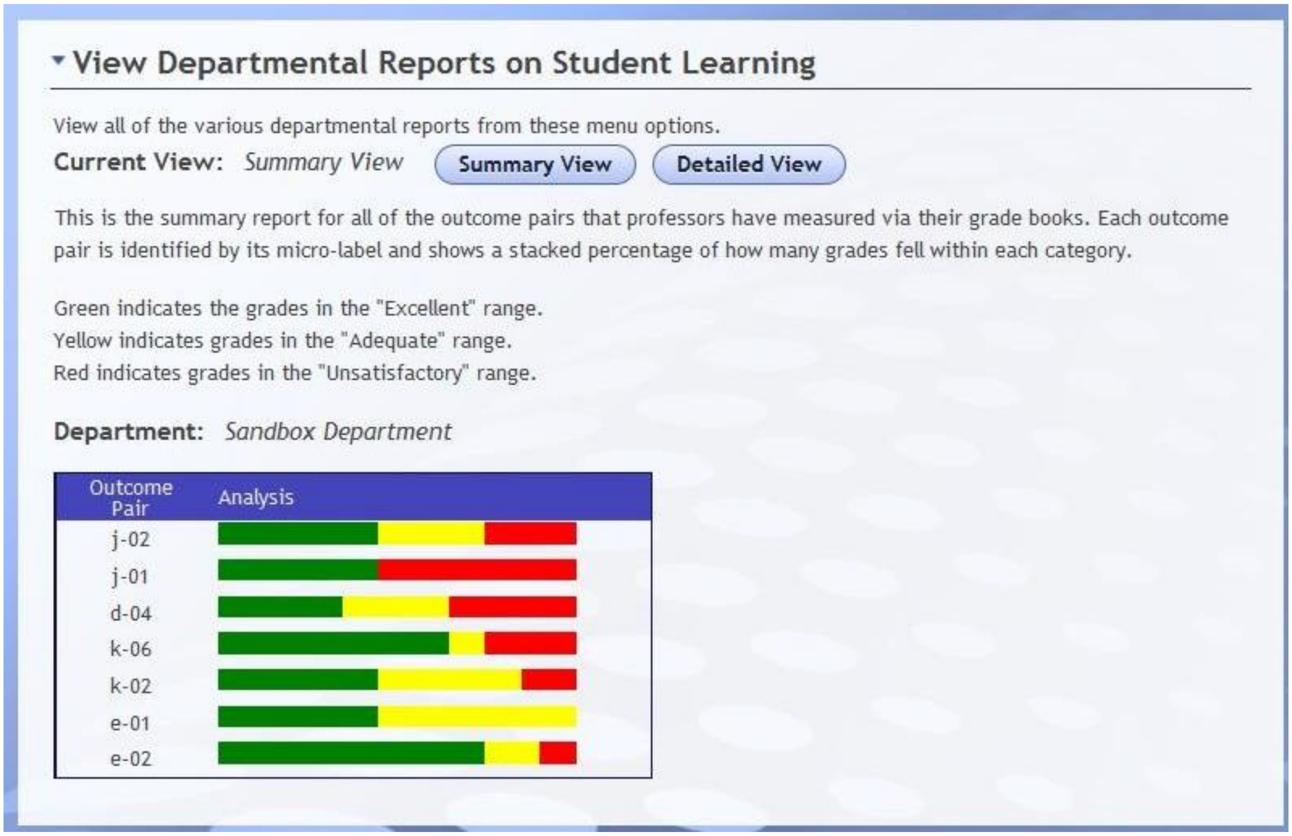

Figure 9: Sample Department Report

While this high-level view is useful, it does have a built-in bias to it: the report does not normalize the scores on a per-student basis. This means that if a single student was measured multiple times for the same outcome, it can skew what the overall performance actually is. Future reports may need to be written to accommodate this bias and give a more accurate representation for how the scores are distributed over all students.

At any point, the user will be able to enter the Detailed View. This detailed view will offer more details for each outcome pair. In addition to seeing the micro-label, a full description of each outcome is shown as well as more information about the actual percentages displayed, the number of scores measured, etc.

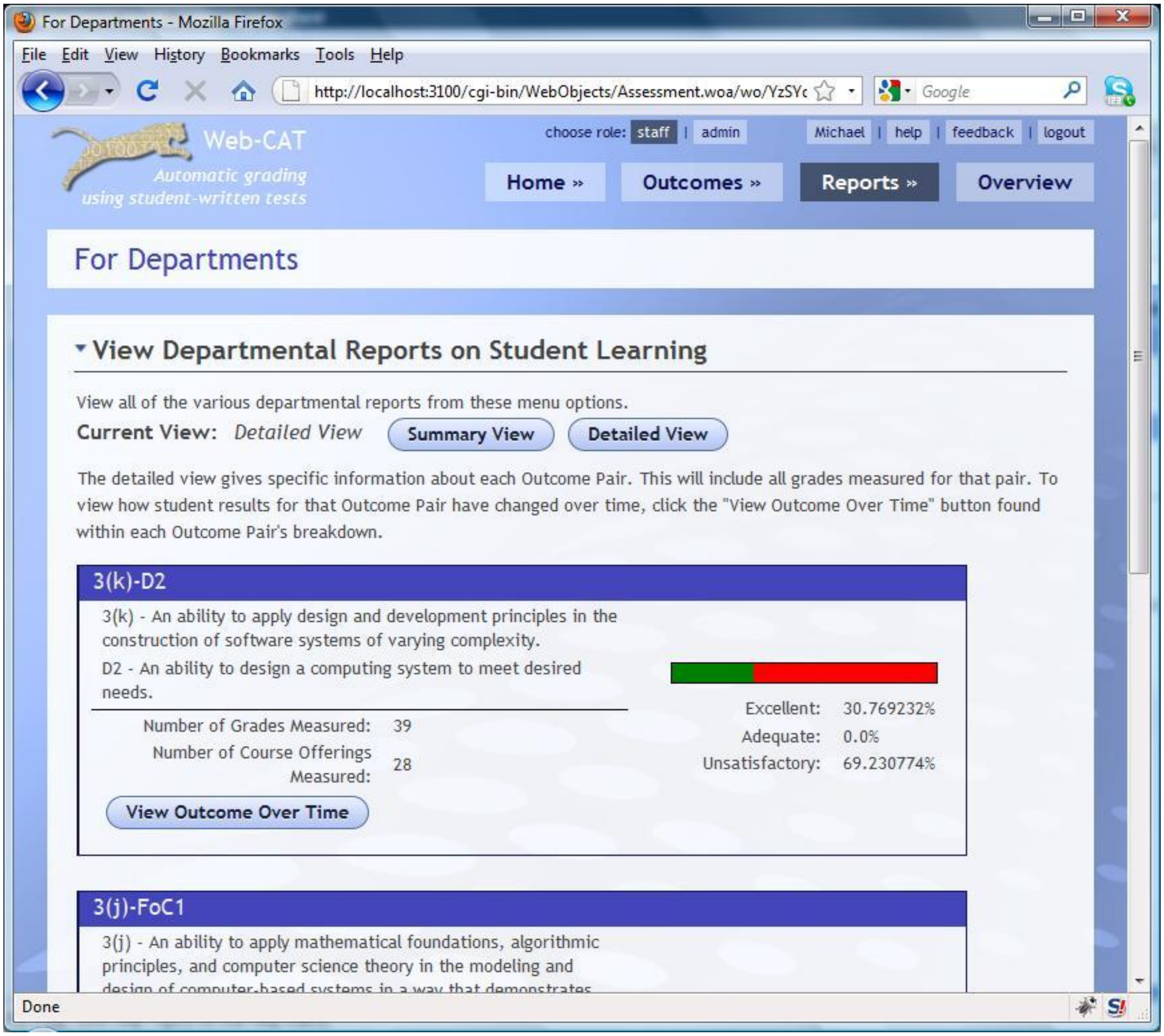

Figure 10: Department Report, Detailed View

This detailed view can be used to get a clearer idea of how well a particular outcome pair is being measured.

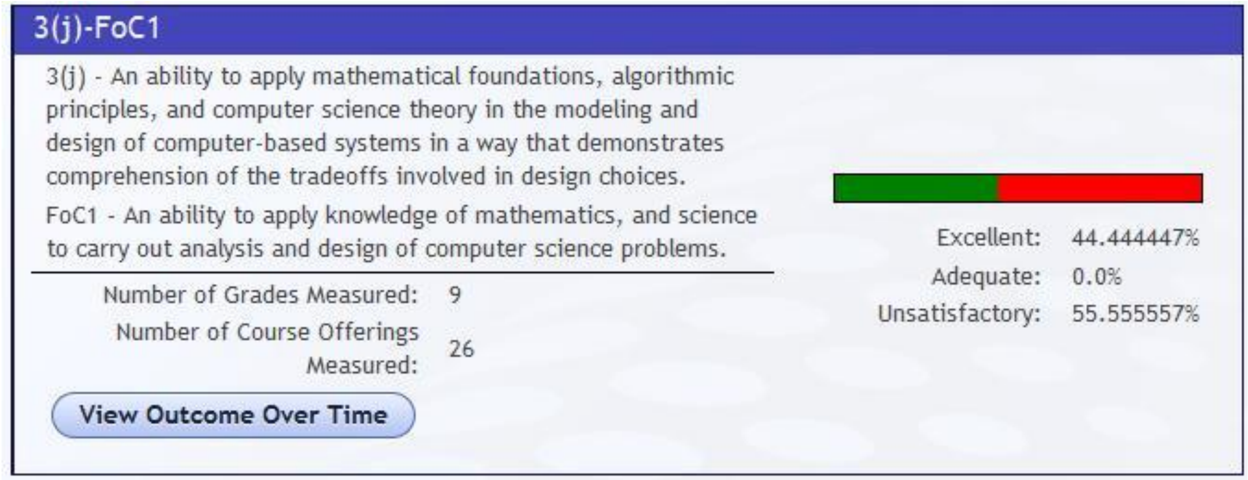

Figure 11: Close up of the Detailed View

The grades are stored in the database as a normalized percent, thus weighting each score equally with all other scores. Since both the raw scores and the normalized score are saved, it is possible to write other reports that will group the measurements by the maximum number of points earned. This will allow for weighted statistics to be viewed. However, due to the arbitrarily large number of methods in which this data can be viewed, we focused primarily on the normalized data.

Other reports offer a similar view of the data based on different search criteria. These differing reports can often have parameters that will allow you to pick a different search term. For example, the report "For Semester" defaults to the most recently completed semester, but allows the user to pick a different semester.

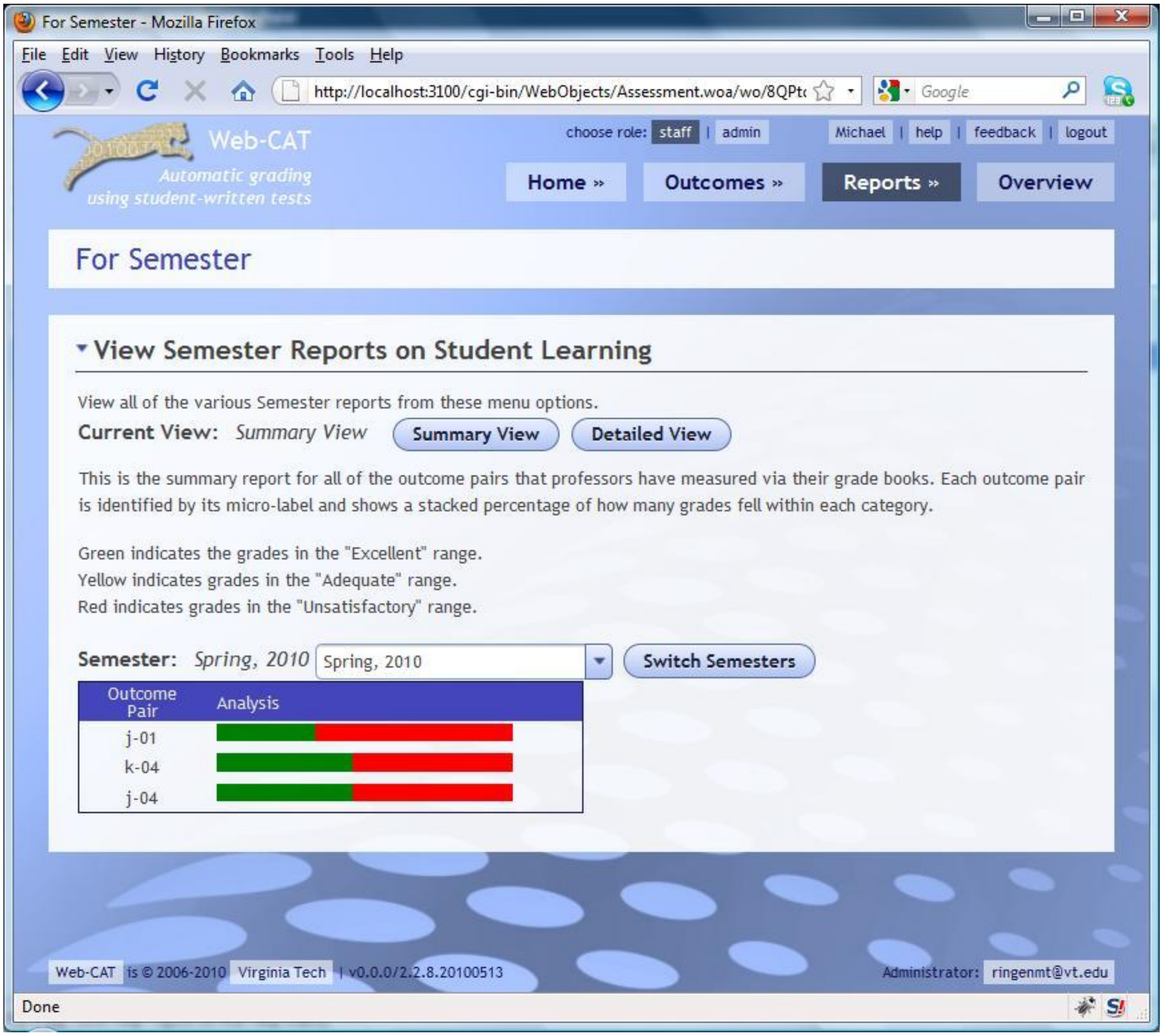

Figure 12: Sample Semester Report

This formatting of the reports page is the planned template for all future reports. It is a clear and concise display that scales well with large numbers of outcomes and measurements.

In addition to the Summary and Detail views, there is future work that plans to add an even more detailed view that will be viewable on the individual outcome pairs in the detailed report view. This highly detailed breakdown is actually a separate report known as the "Outcome Pair-Over-Time" report, which will take the number of scores in each category for a particular Outcome Pair and display how that number changes over time for every semester in which data for that Outcome Pair was collected. Other Outcome Pairs will be selected from a menu allowing the user to view each Outcome Pair's changes over time. This information becomes important for accreditation assessment in that it is a fairly substantial indicator of how well an educational institution is doing in improving its courses and programs.

3.4 Inclusion of Student Program Data

One of the primary problems with this system was the lack of available data that would normally be collected within a CMS. The report generation features are another example where more information pertaining to students outside of the context of a course offering could allow for more detailed analysis of data trends. An example of this would be the separation of data based on whether a student is an under-graduate taking a graduate-level course or vice versa. Another example would be whether a student is taking an "in-major" course or one that is outside his or her discipline. A third piece of information would be to determine which program the student is enrolled in, such as a Bachelors or Masters program.

While the above statistics could be used for the purpose of measuring accreditation standards, there are other statistics that could be valuable for departmental course and program design. Performance of students who transfer in to a program could be separated out and viewed to identify if certain principles are being taught in earlier classes for a program that transfer students may not have as much exposure to prior to entering the program.

Chapter 4: Data Model

4.1 Web-CAT's Data Model

Once the decision to integrate in to the Web-CAT system was made, we began to look at the existing data model. The first important piece of the model was how the courses were modeled. Here is a breakdown of the entities in the model:

User	Any user in the system. Other fields are available in the full data model.
Department	An Academic Institution's college or department.
InstructorCourse	A table mapping which users are acting as instructors for a course section.
CourseStudent	A table mapping which users are enrolled in a course section or offering.
Course	A class belonging to a specific department.
CourseOffering	An offering (section) of a course
Semester	A validation table identifying what a semester is by year, season, and start and end date

Let us begin examining this model by looking at the difference between a Course and a CourseOffering. A Course is created by a Department to teach certain objectives or skills (e.g. CS 101: Introduction to Computer Science). When a department is planning what courses to offer during a given Semester, they create a CourseOffering, commonly referred to in academia as a Section. This allows the department to track information relating to the course for each time it was taught. This is very important in assessment for the analysis of long-term data trends. This information could include such elements as how students who took CourseOffering X in Semester Y fared in subsequent courses or how well students in different offerings of the same course performed under different instructors.

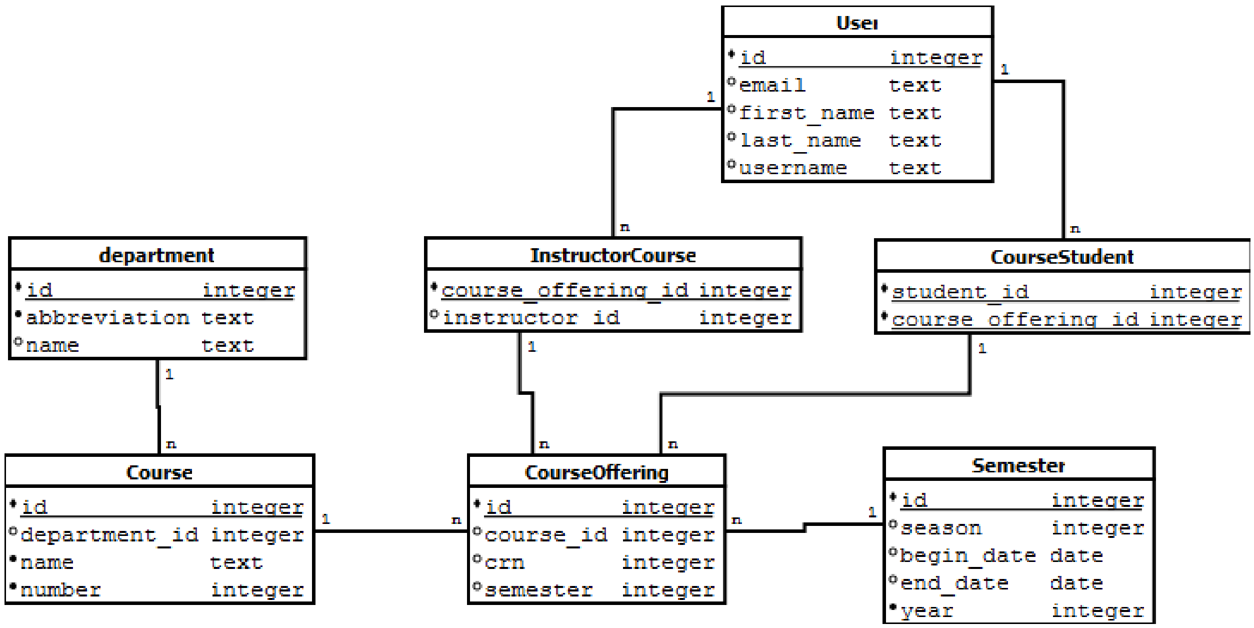

Figure 13: Data Model - User, Courses and Semesters

With the establishment of how a course is offered to the students, we begin to look at how to assess these courses. Web-CAT uses a system of Measures that are applied to each Course. A table called Measure describes what a Measure is, namely the description of a skill or outcome that should be achievable by the students. Some examples of Measures from the Computer Science Department can be seen below.

- Apply knowledge of computing and mathematics expected of introductory computer science majors.
- Create a working program that solves a stated computing problem.
- Understand professional and ethical responsibility in terms of operating systems.

These measures are also given a key phrase in the table so as to group Measures into categories of like topics (e.g. Design and Evaluate, Knowledge of Subdisciplines, Computing and Math, etc.).

When a Measure is created, it can be applied to different course offerings for different courses. An intermediary table called MeasureOfOffering exists to keep track of which Measures are being applied to which CourseOffering. This intermediary table allows each instructor to specify different criteria to determine what constitutes different levels of performance - excellent, adequate, and unsatisfactory. The instructor can then record the percentage of students that met each criterion. The instructor also includes a description of the data used in the measure.

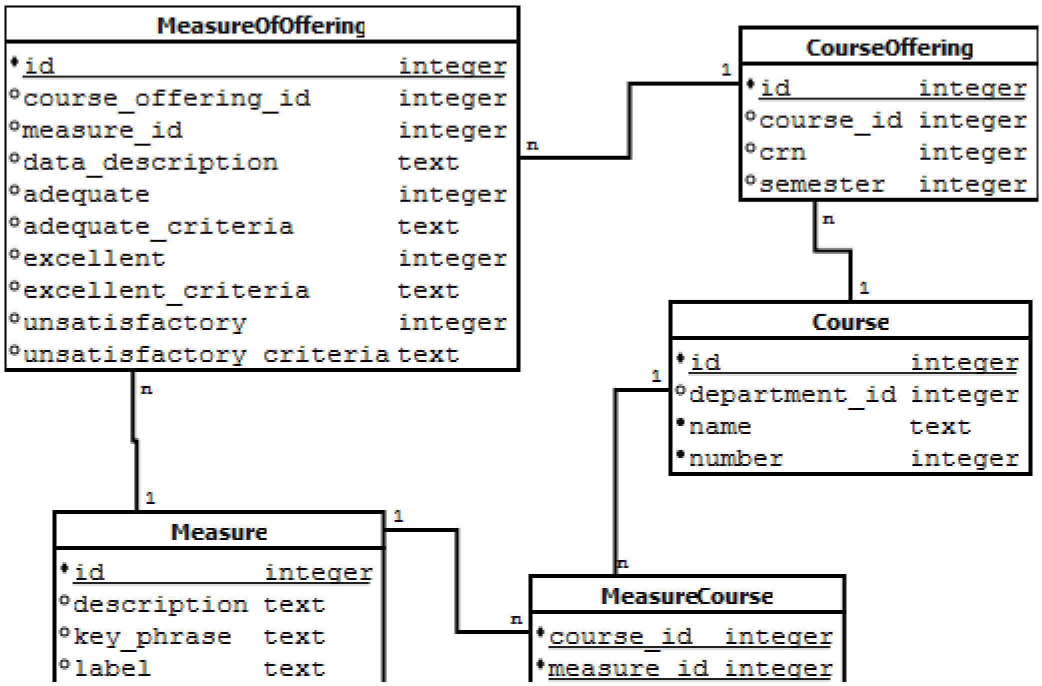

Figure 14: Data Model - Measures

The data used in the MeasureOfOffering could be an exam question, an in-class assignment, homework or just about any other coursework the instructor might offer. This method of collecting course-specific information can be time-consuming and fails to capture data at the level of individual students. However, it does provide quantitative data on how students as a whole handled a specific offering of a course.

In addition to the quantitative assessment, instructors are given the ability to record their own reflections on how they taught the course. This information is held in the FacultyReflection table.

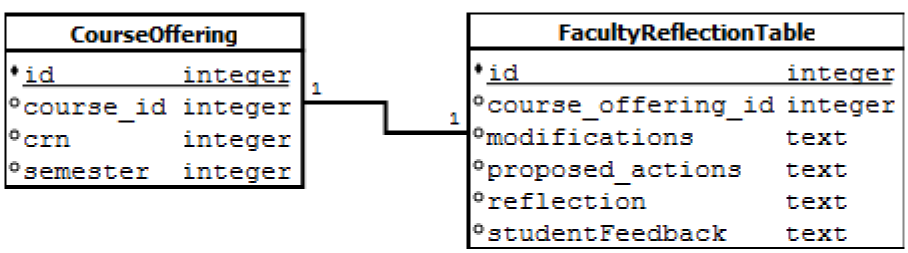

Figure 15: Data Model - Faculty Reflections

The table allows the professor to record any modifications he made to the course, proposed actions for future changes, his or her reflections on the course and any student feedback they should wish to include. This table lets administrators compare how different sections of a course were taught and compare the effectiveness of different modifications or teaching styles. The proposed actions field can be a valuable tool for designing the curriculum. Patterns and common themes can be discerned, analyzed and implemented resulting in a stronger curriculum.

So far, the data model has been describing data internal to the academic institution. While the assessment capabilities of these tables are useful for internal course audits, we have no way of tying this data to program or external outcomes. With our understanding of the current data model, we'll begin expanding this to map learning outcomes for both accreditation and program assessment.

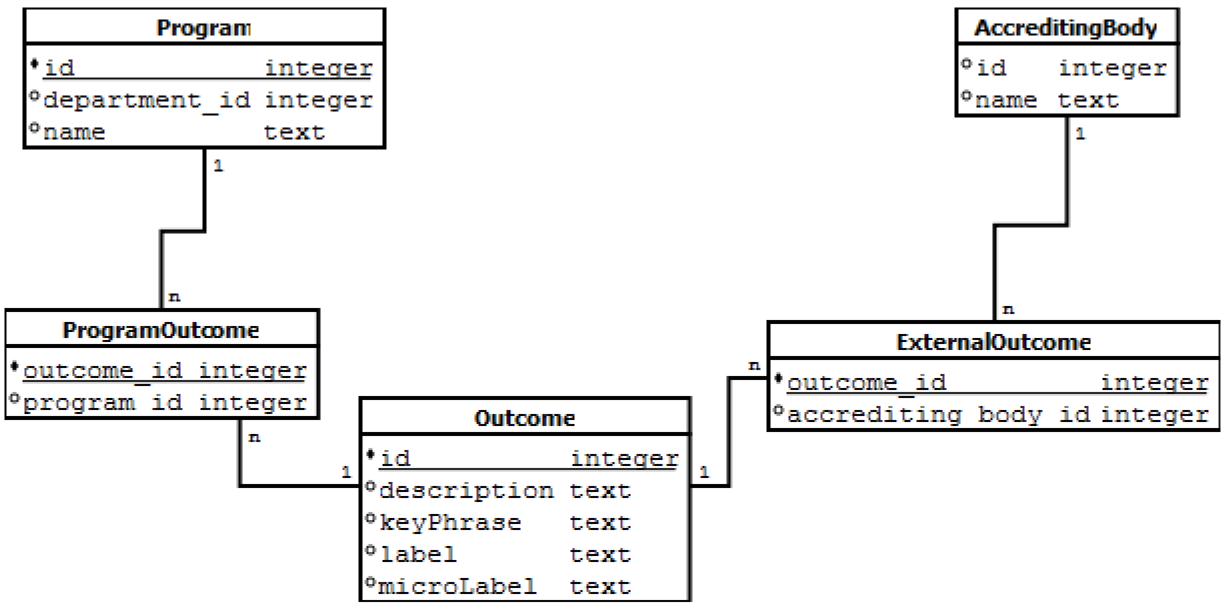

Figure 16: Data Model - Program, Accrediting Body, and Outcomes

All outcomes are stored in the Outcome table. Each Departmental program will have some number of outcomes associated with it, as will each Accrediting Body. The two tables, ProgramOutcome and ExternalOutcome establish the relationships between each entity and its respective outcomes. However, this basic structure leaves out several key pieces of information, namely:

1. Programs, Outcomes and Accreditation standards are sure to change over time. How can we know what outcomes were used for a given time period?
2. What, if any, relationship exists between the AccreditingBody's outcomes and the departmental program's outcomes?

We now expand on this data model to address these two issues. To solve the first issue, we begin by arranging each grouping of program outcomes in to sets and then give this set a time period that it is related to.

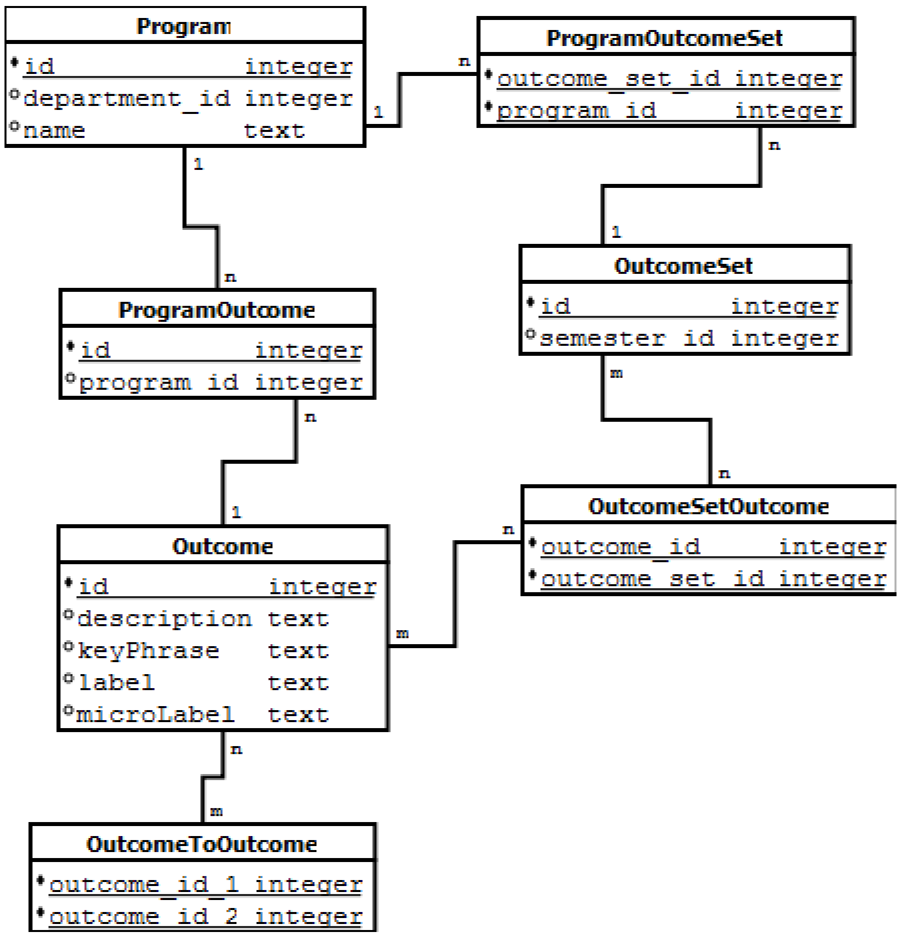

Figure 17: Data Model - Program Outcomes

We are already familiar with the first three entities in the left column. The last table, the OutcomeToOutcome allows us to form relationships between outcomes. It is the three columns on the right that establish the solution to the first problem raised above. Each Program will have a ProgramOutcomeSet. This ProgramOutcomeSet will contain one or more OutcomeSets. These OutcomeSets record which semester the OutcomeSet was active, thus giving us an idea of over the course of a Program, we know which OutcomeSets were active. To populate the OutcomeSet with Outcomes , we create the OutcomeSetOutcome table. This table simply holds the id of each outcome that is associated with a set. If an Outcome is to be used over the course of multiple semesters or for multiple Programs, it will have multiple records in the OutcomeSetOutcome, OutcomeSet, and

ProgramOutcomeSet tables, thus keeping the record-keeping normalized. By querying on the Outcome table, information for each outcome can be retrieved about which programs it was used for, and when it was first used and last used for that program.

This same structure exists for the Accrediting Body outcomes, as seen below:

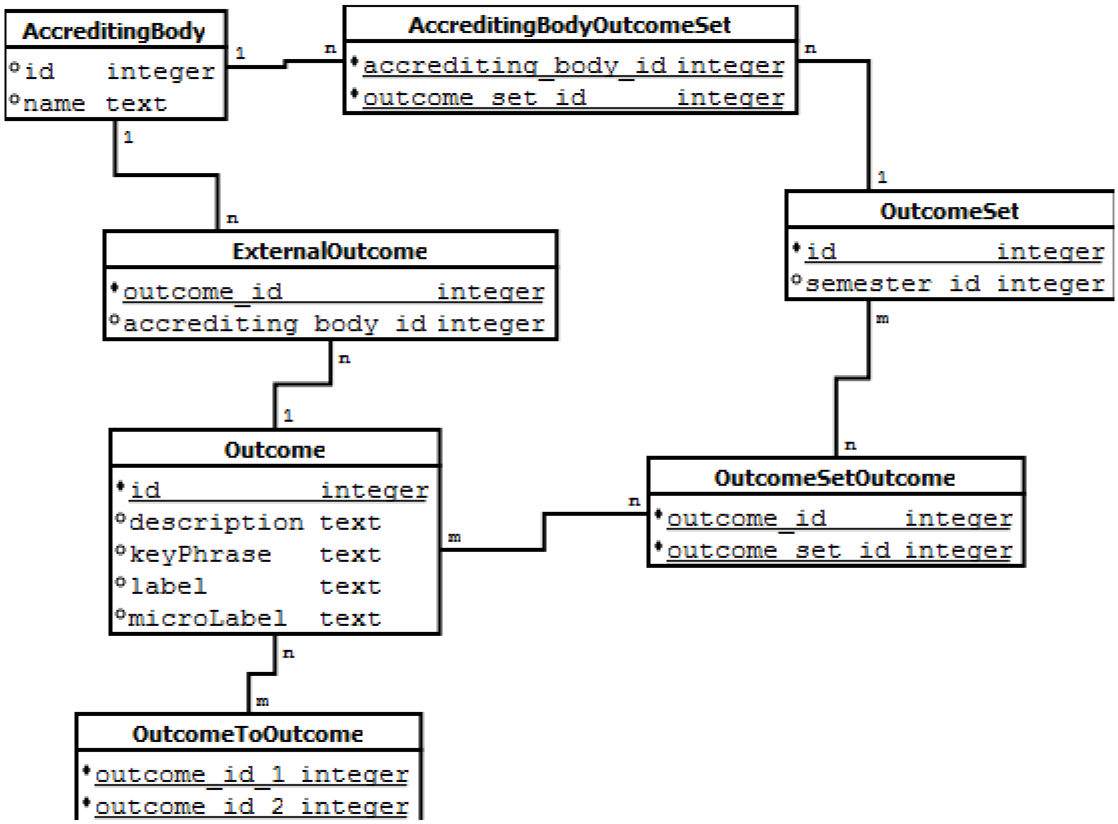

Figure 18: Data Model - Accrediting Outcomes

The structure of the diagram seen here mirrors that of the Program Outcomes save that it now uses the tables specific to Accreditation. ExternalOutcomes replaces the ProgramOutcomes and the AccreditingBodyOutcomeSet replaces the ProgramOutcomeSet.

We have now adequately addressed the problem of knowing when an Outcome was used and what happens when the outcomes for a given time period change. There is one more concern about the problem of active Outcomes. Let us use an example to highlight the problem. The computer Science department decides that it needs to re-design its Bachelor of Science program for Computer Science. All existing students will be grandfathered in, staying in the original program, but all new students will be enrolled in the new program. If we assume that the newly designed program will use several outcomes that the previous program did not, then we must consider what will happen when students enrolled in each program enroll in the same course offering together. Students will either be assessed based on outcomes that don't pertain to them, or the assessment process will be expanded to allow the instructors to evaluate each student according to the outcomes pertinent to them. This latter option is fairly grand in scale and could be cost-prohibitive. A similar problem will occur when students take an out-of-major class. We will discuss this more in the future work section, but it is important to be made aware of this problem from now.

Let us now address the second issue mentioned above, namely how to relate program outcomes to external outcomes. The Computer Science department has devised its program outcomes to be similar to the external outcomes. We are therefore able to use a pairing schema that will match each program outcome to a corresponding external outcome. This is done on a per course basis, allowing for each outcome to be mapped multiple times based on what the focus of a course is.

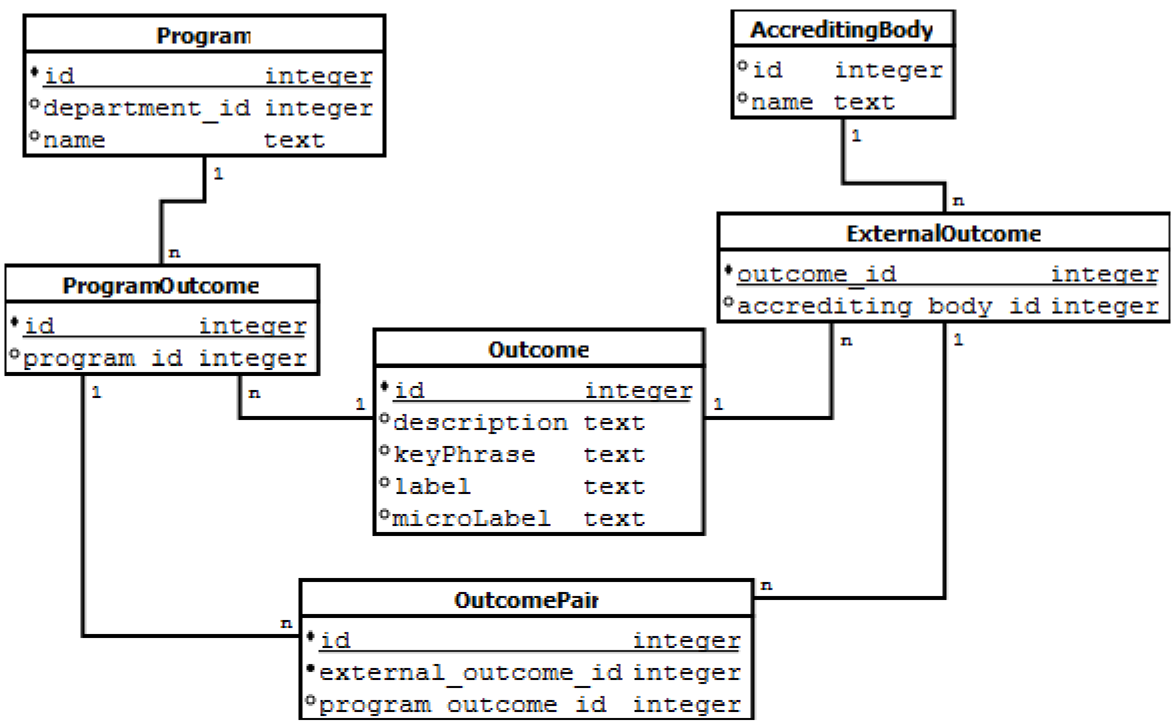

Figure 19: Data Model - Outcome Pairs

This single table addition serves the purpose of establishing the relationship between program and external outcomes. Within the Virginia Tech Computer Science department the establishing of these relationships is handled by members of the assessment committee and administrative staffing. These Outcome Pairs are then used for accreditation assessment.

Once we have established these pairs, we look again at the Measures we have to assess them and begin to bring the data model together.

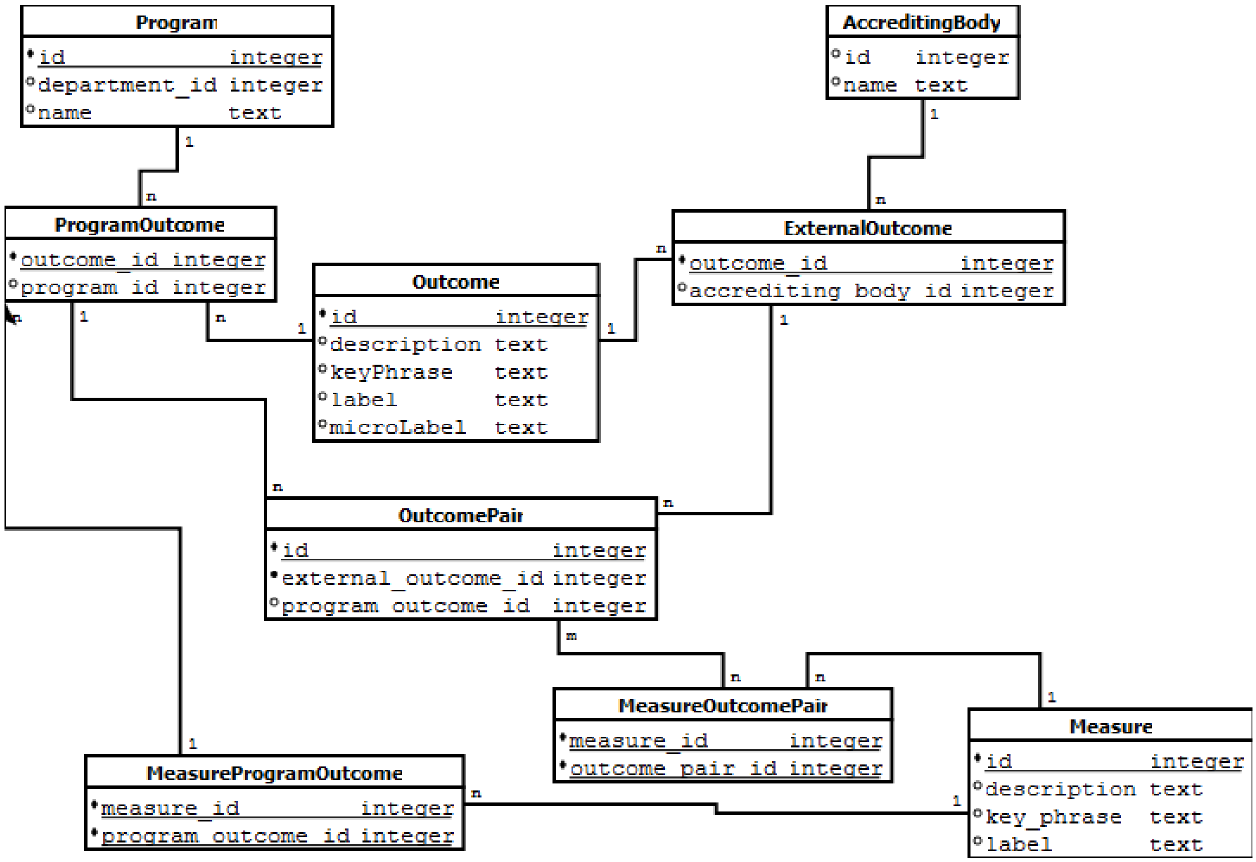

Figure 20: Data Model - Outcomes and Measures

ProgramOutcomes can be tied directly to a Measure for our own internal assessment purposes. However, it is the OutcomePair that is assessed and evaluated for the accreditation Measures. While this does create a slight redundancy in the Measuring of program outcomes, it also allows for Measures to be tied very specifically to OutcomePairs. By pairing the Outcomes together, it is not the tying of unrelated elements and thus causing a potential loss of specificity; rather it is a complimentary pairing that allows for a further refinement or focusing of what the purpose of the outcome is and how it should be measured.

This brings us to the full data model we began working with. From here, we begin to integrate our own entities in to the data model.

4.2 Evolution of the Data Model

We wanted to add to the data model our idea of recording and measuring student data and using that as a basis for accreditation assessment. We began this by creating a model for how to record an assignment. The first iteration of the data model had separate entities for storing a coursework and the questions belonging to that coursework. Each coursework belongs to an offering of a course. Outcomes of whatever type mapped directly to a question. The simplified diagram can be seen in figure 21.

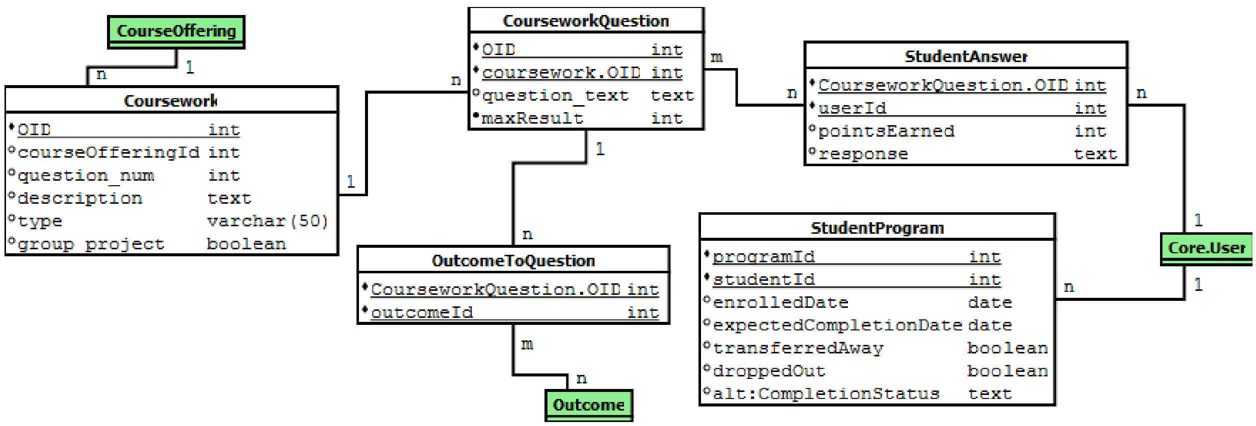

Figure 21: Data Model Addition - Initial Coursework and Outcomes

Several problems immediately became evident. First, all of the information found in the StudentProgram entity was not readily accessible for entry in to the system. This information can be found or derived from the Banner system used by Virginia Tech. However, it would take an administrator for the Computer Science department time and significant effort to retrieve this information from Banner, format it for our system and then re-enter it. Looking at the scope of the project, this data seemed auxiliary to the actual focus of assessment. This student meta-data is most useful for course analysis internal to the program. This internal course analysis can be used to improve course and program design as well as offer advice to students on what offerings they should take and when based on the performance of other students in similar circumstances. Since it became evident that this could be safely removed from the data model and the design and implementation of this module deferred to a future date, it was removed. See the Future Work section for project ideas regarding this.

The next decision to make in the data model was how to record the assignment information. The separation of Coursework and Questions in to two distinct entities has benefits and drawbacks. Of high importance is the preservation of the coursework's integrity. By using two entities, we can keep the logical grouping of questions that belong to the same assignment. Outcomes could then be mapped at the coursework or question level. By recording the mapping at the question level, data mining can then determine patterns for outcome grouping. If certain outcomes show up in several assignments, analysis can determine the relative performance of students when those same outcomes are paired with other outcomes. Also of relative importance is the availability of information pertaining to the coursework itself. Information such as assignment length, assignment type, group projects, in-class assignments versus take-home and other similar elements can then be mined for possible impact or effectiveness in learning for a particular outcome. These benefits though come at a high cost. To obtain this information from the gradebook, additional data elements would have to be stored that is not likely to be found in the current gradebooks that professors typically keep. It would also require a less flexible gradebook format, which would further impact professors and the amount of time they would have to spend accommodating our system. This conundrum again highlights the problem of course assessment overlapping with course management. In a course management system, defining and recording meta-information on coursework would be appropriate and would dovetail neatly in to the existing workflow for such a system.

We then looked at a second possibility for storing coursework information. I combined the coursework and question information in a single table called "coursework." (fig. 22) This coursework table stores only the basic information about the coursework - which course offering it was assigned in, what Outcome it maps to and how many points it was worth. This "bare-bones" approach actually allows us to treat anything in a professor's gradebook identically without having to worry about what it actually is - a coursework or a question, an exam or a group project. This single-entity design misses a lot of the meta-information describing the coursework as well as losing the logical grouping of questions into a single assignment. However, this separates the assessment system from the role of the course management system; there is no overlap or redundancy in the data entry or record-keeping that professors must perform.

Coursework	
•OID	int
°courseOfferingId	int
°max points	int

Figure 22: Data Model - Coursework

After weighing the potential pros and cons of the afore-mentioned designs in relation to our focus and goals, we opted for a single coursework table. Given that our primary goal for this first attempt was ease of use, we didn't want to force the users to significantly alter their methods for keeping their gradebooks. While some professors may keep their grades on a per-question basis, others may only record the points earned per-assignment. Our coursework table will accept both means and record them identically.

The corresponding table called StudentAnswer records exactly that. It stores the student, the coursework, and the points and percent earned. There was very little other information needed in this table. While the original version of the table did not contain the "percent earned" column, it was added upon reflection of how the reports were generated. For large data sets, the normalization of data was slowing the response time of the system down when certain reports were being generated. By calculating and storing the percentage upon the creation of the studentanswer row, the cost of normalizing the data is now negligible.

These two tables, Coursework and StudentAnswer, adequately handle the storing of information normally found in a professor's gradebook. The final entity in the data model is the storing of outcomes. We focused on two types of outcomes - External Outcomes and Program Outcomes. External Outcomes are Outcomes generated by an Accrediting Organization. Program Outcomes are outcomes generated by a department with regards to a specific program (e.g. Bachelors of Science in Computer Science or Masters of Science in Computer Science). For each course offering, the Computer Science department pairs a Program Outcome to an External Outcome and gives a list of such pairs to the instructors for that course. It is these Outcome pairs that get mapped to the Coursework objects. The data model already contained the OutcomePairs table so we simply added a relationship to the Coursework table. Other departments or programs may not use this model of pairing outcomes together. With this in mind, the ER diagram simply shows an entity labeled "Outcomes", conveniently abstracting away from the individual means by which Outcomes are recorded and stored. The final data model can be seen in Figure 23.

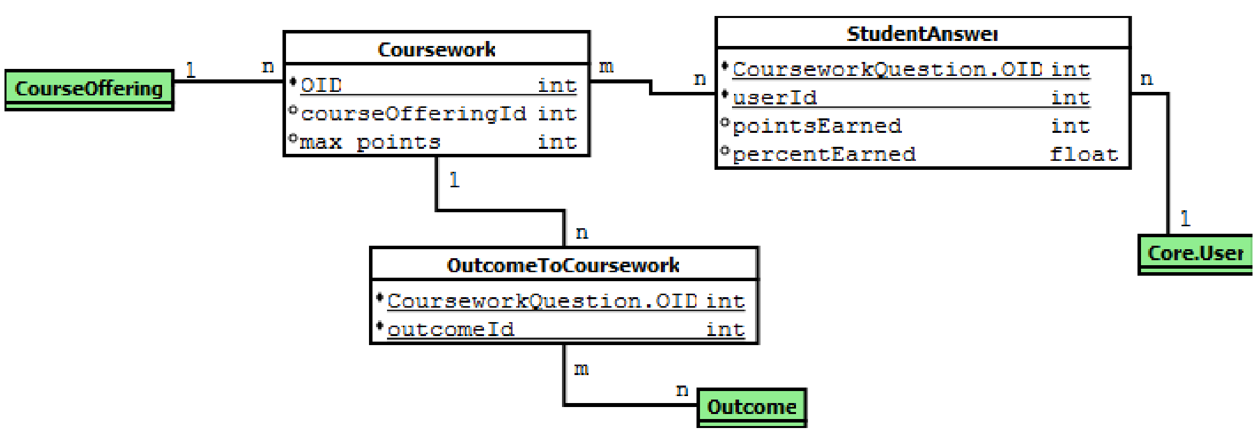

Figure 23: Data Model - Coursework, Student Answers, and Outcome Mappings

4.3 Benefits of Data Model

This design strongly accommodates many of the goals we sought to achieve with this application. First and foremost, it stores the direct relationship between the grade a student earns and the outcomes to which that grade relates. It is a quantifiable measurement of how well a student is achieving a particular outcome.

Second, the data model allows for the tracking of outcomes over time. It allows for this tracking from multiple perspectives. An individual student's progress can be analyzed to see how that student's performance has changed from semester to semester (i.e. see whether they are improving in a particular outcome). The data model also allows for a course to be analyzed, helping to determine course quality or professor quality. A departmental program can also be analyzed, seeing if there is a correlation to student learning in regards to what order courses are taken.

Finally, the data model is flexible enough to accommodate professors and the various behaviors that will be employed while using the system. The generic coursework table allows the system to record any number or frequency of measurements using any point scale without adding a bias to the recording.

4.4 Problems with the Data Model

The major drawback to the data model is a result of the major problem discussed throughout this project; namely that it is built as a stand-alone application instead of being integrated with a Course Management System (CMS). Several decisions had to be made as to where to cut off information gathering so as not to overlap with a CMS and thus cause redundancies in either the data collected or in the workflow of a professor. Given that one of the main goals of this system is to have a minimal impact on the workflow or time commitment of the end-users, this duplication of work had to be avoided.

In view of this, the data model is designed as a "fire-and-forget" system. When a professor uploads a gradebook, it does no verification to see if rows for that course number already exist. This prevents incremental or multiple uploads of the same spreadsheet. More specifically, when a professor uploads a spreadsheet that has been previously stored in the system, it will insert duplicate rows in to the data model for each student record that was present the first time the spreadsheet was uploaded. A simple immediate fix that will be added, time permitting, is that for each unique course number in a spreadsheet, all previous rows for that course number will be deleted from both the coursework and student answer tables. This will allow for multiple uploads during a given semester, allowing a professor to analyze trends within a class. Problem topics could be easily identified and rectified within a semester, should a professor upload and see that all students are having trouble with assignments relating to a particular outcome. Alternatively, if a single student is struggling with a topic, the reports would make this evident and the professor can offer additional help to that student on topics relating to that outcome.

Finally, the data model does not include extra information about the student. This information could include what program they are enrolled in and thus whether they are attending a class outside of their primary discipline, when their enrollment date and expected graduation date are, and whether they successfully completed the program (as opposed to dropping out or transferring away). This extra information could be of benefit when preparing data for the accreditation institution as well as being beneficial within a department. The primary problem with gathering this information is that it is purely administrative and is not readily available outside of a CMS.

Chapter 5: Conclusion

5.1 Summary of Problems and Solutions

The primary problem we are seeking to address is the lack of direct quantitative student data being measured in the current accreditation process. Secondarily, we are trying to provide an efficient means of collecting this data. Finally, we are trying to address the issue of making accreditation a continuous process as opposed to a series of discreet events. To this end our project was moderately successful with good staging to be expanded and refined and further meet these goals.

For the primary problem of collecting quantitative student data, our system was designed to directly address this issue. Professors have the option of providing student scores on a per-question or a per-assignment basis. These scores are then directly mapped to the accreditation outcomes and program outcomes relevant to the course, thus allowing for direct quantitative support for conclusions about how the program is changing over time, either improving or worsening.

The second task was to find a means of effectively collecting this data. Many professors do already keep their gradebooks in electronic format, with spreadsheets being the most common form of record-keeping. We have added one step to this process, asking professors to map their grades to the outcomes. From there, only one other task - that of uploading the gradebook to the system - has been added to their workflow. We believe this impact on professors is minimal, giving us a greater benefit to cost ratio than what other quantitative assessment solutions have offered in the past.

The final task of making accreditation assessment more of a continuous process is the only problem that we have not fully addressed. In the past, assessment would happen on a cycle spanning years. With this system, we have closed that gap down to the time frame of a single semester. However, we believe that this could be improved upon with improvements to the nature of the system. This will be discussed more in the future work section.

5.2 Future Work

There are several key areas for improvement and research with the Web-CAT module. Before improvements can be made, however, we must first evaluate the tool itself.

5.2.1 Evaluating the Web-CAT Module

While the tool has been completed it has not yet been evaluated by its intended audiences. There are two aspects that need to be evaluated. The first is to introduce it to professors and ensure it meets the criteria of efficient use. Professors should be introduced to the tool before the beginning of a semester, explaining the purpose of the tool as well as the acceptable formats for a gradebook spreadsheet. This will give professors time during the planning phase of a course offering to evaluate their planned coursework, mapping the learning outcomes as the coursework is being designed. It is possible that some benefit to course design could be realized during this phase of the evaluation should a professor find a particular coursework that does not directly map to any learning outcomes. At the end of the semester, professors will be expected to use the tool to upload their gradebooks as described earlier in this paper. After this task is completed, a short subjective survey can be filled out to ascertain whether the tool met the stated goals of fitting in to the professor's existing workflow, whether the reports are useful to them, and what suggestions, if any, they would have for improving the tool.

The second aspect that needs to be evaluated is whether the data that is collected by the professors and presented in the reports to ABET actually conveys a useful quantitative measurement of the effectiveness of the program being assessed. Discussions with ABET can be used to ascertain what other data elements they would like to see collected as well as other reports generated from the data.

5.2.2 Course Management System Integration

Throughout this paper, I've discussed the overlap of assessment with a course management system (CMS) and the problems of cutting off or limiting functionality based on this overlap. This is a broad area for future work as there are several directions that could reasonably be taken. This integration could either take the form of an API interface with an open-source CMS or as a batch file transfer from the CMS to Web-CAT or vice versa. Giving professors the ability to upload or synchronize their gradebook during a semester would allow for evaluation of student performance during the semester itself. It is conceivable that this mid-semester course evaluation would allow for potential problems with students to be identified early enough that extra support could be offered. It could also show trends where a certain skill or skills (as opposed to topics) aren't being fully realized.

Additionally, integrating with a CMS could further ease the burden imposed on professors by eliminating a step in the process. Professors who choose to store their gradebook within the CMS itself would no longer need to go through the step of uploading it to the system. The benefits mentioned above of continuous synchronization and evaluation of the gradebook could also be realized.

5.2.3 Registrar Data Integration

Another key point of integration is with the registrar system. As mentioned earlier, correlating the results of coursework results to student registrar information can help identify potential problem areas with future students. The registrar could include such information as when the student was enrolled, expected graduation, course history, major/minor, undergraduate or graduate, transfer status, and others. While some of this information could be useful for assessment (such as separating or eliminating the variables of non-majors from accreditation assessment), it is speculated that this information could be very important for internal purposes, allowing the educational institution to audit itself, improving course and program design based on these criteria. It will also help tailor course selection and ordering of courses taken for majors and non-majors alike based on trend data.

5.2.4 Evaluating Coursework and Professors

The richness of the data captured allows for two other areas of assessment. Using the principles and methods of Item Response Theory, it is possible to evaluate the quality of any given coursework. Just as IRT is used to develop questions on standardized exams, it can be used here to identify especially high or poor quality coursework, allowing for refinement and improvement.

Similarly, IRT can be used to determine if a particular professor is more suited to teaching certain classes than others. By evaluating the results of a series of course offerings, trends can appear about the effectiveness of a professors teaching for different classes, allowing departmental administrators to further refine and improve their courses by assigning more suitable professors to teach various classes. It should be noted that in the Literature Review, several instances occur where professors expressed dis-like for a given assessment system due to the fact that they felt as though they were being evaluated as opposed to the course or the program. This sensitive topic must be approached from a positive aspect to help ensure continuing cooperation and collaboration with professors.

5.2.5 Expanding the Data Model

Another area of possible research is to evaluate the benefits of further refining the data model. The original model contained questions and coursework in separate tables. In doing so, a significant amount of meta-data can be recorded. It is not known how useful this would be for accreditation purposes, but again, for internal assessment the data mined could be used to refine course and program design. This meta-data was mentioned earlier in the report, but could include such information as assignment type, in-class versus take-home, group work vs. solo, long-term project vs. short-term, and grading rubrics. The key problem to tackle with this is finding a way for professors to record this information in a manner the system could recognize without completely changing their grade-keeping methods. The current system allows for some flexibility in the gradebook with a stated goal of increasing grade-keeping flexibility as the recognition algorithms are improved. However, some potential solutions for this problem could be that in a single-sheet gradebook, assignments could be color-coded to go together, or a merged field above all the relevant columns to indicate grouping. A study would most likely need to be done in order to determine the usability and practicality of various recording methods, as well as evaluating these changes in terms of changes to the Web-CAT module.

While other future work could be done, I believe these are the next major areas that should be considered to further the principles of assessment, both for internal and for accreditation purposes. Other improvements to the system need to be done that don't necessarily qualify as research topics. The system currently offers only a minimal number of reports. The research done so far focuses primarily on the means of collecting the student data from the professors without focusing on the specifics of how the student data will be viewed. While this paper speculates extensively on the various directions in which the data could be used in reports, no research has been done on what would prove most valuable and no reports exist to fully organize this data.

Another somewhat minor topic would be to improve the spreadsheet identification algorithms, allowing for greater flexibility in how professor's keep their gradebooks. While the current system does offer a fairly standard set of options, it is fairly restrictive. The methods weren't expanded upon because of the limited number of gradebooks analyzed prior to the creation of the project. As more and more professors begin to use the system, we'll have access to more gradebooks and will be able to adjust and expand the algorithms to better suit the variety and personal preference of professors.

5.3 Summary

The Web-CAT module provides a significant increase over other tools in the amount of student data collected for assessment purposes. The time and effort that professors need to devote to assessment is also minimal compared to other tools, with much greater benefit to all involved - students, professors, administrators and accreditation institutions. With this targeted focus on actual student learning, accreditation institutions such as ABET can accurately gauge the quality of education that a program offers. Administrators and professors can use the tool to improve the curriculum for a program as well as offer advice on course selection to individual students based on trend data. It also alleviates or avoids the common pitfalls that other assessment methods incur. The assessment Web-CAT module negates the idea of "teaching to a standardized test". It also removes the qualitative folk pedagogy that is often the basis for or a major component assessment in higher education. Overall, while the system is not perfect, it takes an important step towards quantifying assessment of student learning for accreditation purposes.

References

[1] (ABET, Inc., the recognized accreditor for college and university programs in applied sicence, computing, engineering and technology., 2010)

[2] Aasheim, C., Gowan, A. and Reichgelt, H. An IT program assessment process *Proceedings of the 8th ACM SIGITE conference on Information technology education*, ACM, Destin, Florida, USA, 2007.

[3] Abunawass, A., Lloyd, W. and Rudolph, E. COMPASS: a CS program assessment project. *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*, ACM, Leeds, United Kingdom, 2004.

[4] Ala-Mutka, K.M. A Survey of Automated Assessment Approaches for Programming Assignments *Computer Science Education*, 15 (2). 83 - 102.

[5] Blaha, K.D. and Murphy, L.C. Targeting assessment: how to hit the bull's eye. J. Comput. Small Coll., 17 (2). 112-121.

[6] Blandford, D.K. and Hwang, D.J. Five easy but effective assessment methods. SIGCSE Bull., 35 (1). 41-44.

[7] Booth, L. A database to promote continuous program improvement. *Proceedings of the 7th conference on Information technology education*. Minneapolis, Minnesota, USA, ACM: 83-88. 2006.

[8] Booth, L., Preston, J. and Qu, J. Continuous program improvement: a project to automate record-keeping for accreditation *Proceedings of the 8th ACM SIGITE conference on Information technology education*, ACM, Destin, Florida, USA, 2007.

[9] Booth, T. and Miller, R.E. Computer Science Program Accreditation: the first-year activities of the computing sciences accreditation board. Commun. ACM, 30 (5). 376-388.

[10] Engel, G.L., Cain, T., Dalphin, J.F., Davida, G., Frederick, T.J., Gibbs, N.E., Lidtke, D.K. and Mulder, M.C. Accreditation in computer science *Proceedings of the fifteenth SIGCSE technical symposium on Computer science education*, ACM. 146. 1984.

[11] Rogers, Gloria. The Language of Assessment: Humpty Dumpty Had a Great Fall. Communications Link, published at www.abet.org

[12] Gowan, A., B. MacDonald, et al. A configurable assessment information system. *Proceedings of the 7th conference on Information technology education.* Minneapolis, Minnesota, USA, ACM: 77-82. 2006

[13] Hankins, J.A. and Parker, B.C. The impact of accreditation on CS1 and CS2 *Proceedings of the 33rd annual on Southeast regional conference*, ACM, Clemson, South Carolina, 1995.

[14] Herbert E. Longenecker, J. and Feinstein, D.L. Development of assessment for undergraduate programs of information technology, and certification for program graduates *Proceedings of the 6th conference on Information technology education*, ACM, Newark, NJ, USA, 2005.

[15] Higgins, C., Symeonidis, P. and Tsintsifas, A. The marking system for CourseMaster *Proceedings of the 7th annual conference on Innovation and technology in computer science education*, ACM, Aarhus, Denmark, 2002.

[16] Holden, E., Assessment of an introductory database course: a case study. in, (2008), ACM New York, NY, USA, 131-138.

[17] Jones, L.G. and Price, A.L. Changes in computer science accreditation. Commun. ACM, 45 (8). 99-103.

[18] Leach, R.J. Analysis of ABET accreditation as a software process *Proceedings of the 13th annual conference on Innovation and technology in computer science education*, ACM, Madrid, Spain, 2008.

[19] Maxim, B.R. Closing the loop: assessment and accreditation. J. Comput. Small Coll., 20 (1). 7-18.

[20] Pardo, A. A multi-agent platform for automatic assignment management *Proceedings of the 7th annual conference on Innovation and technology in computer science education*, ACM, Aarhus, Denmark, 2002.

[21] Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M. and Paterson, J. A survey of literature on the teaching of introductory programming *Working group reports on ITiCSE on Innovation and technology in computer science education*, ACM, Dundee, Scotland, 2007.

[22] Sanders, K.E. and McCartney, R. Program assessment tools in computer science: a report from the trenches *Proceedings of the 34th SIGCSE technical symposium on Computer science education*, ACM, Reno, Navada, USA, 2003.

[23] Taylor, H.G. and Martin, C.D. The impact of new accreditation and certification standards for secondary computer science teachers on university computer science departments *Proceedings of the twenty-third SIGCSE technical symposium on Computer science education*, ACM, Kansas City, Missouri, United States, 1992.

[24] Winters, T. and Payne, T. What do students know?: an outcomes-based assessment system *Proceedings of the first international workshop on Computing education research*, ACM, Seattle, WA, USA, 2005.

[25] Yang, K.-p. and Beaubouef, T. Automatic generation of web survey for assessment purposes in computer science. *SIGCSE Bull.*, 40 (4). 78-82

[26] Yue, K.-B. (2007). "Effective course-based learning outcome assessment for ABET accreditation of computing programs." J. Comput. Small Coll. 22(4): 252-259.

[27] Bailie, F. K., A. M. Abunawass, et al. "Guidelines and suggestions for ABET accreditation." J. Comput. Small Coll. 21(2): 83-85. 2005.