# AC 2012-3216: A PRACTICAL AND COMPREHENSIVE APPROACH OF ASSESSING ABET OUTCOME ACHIEVEMENT IN COMPUTER SCIENCE AND COMPUTER ENGINEERING

**Dr. David Wilczynski, University of Southern California**

David Wilczynski has a long history at USC. He was the first Ph.D. graduate from theUSC Information Science Institute in 1975, where some of the initial work on Arpanet was done. His research specialty at the time was in knowledge representation. In 1984, he left USC for almost 20 years to be an entrepreneur. Most of his work was in manufacturing, both in Detroit and Japan. During that time, he worked on programming real-time systems using an agent methodology, which he now teaches in his CSCI 201 class. He returned to USC in 2002 to teach full-time. Mostly, he worries about how to make undergraduate engineering students more professional. Once a tennis player, he is now trying to become a golfer. Bridge, cooking, and his family take the rest of his time.

**Dr. Gisele Ragusa, University of Southern California**

Gisele Ragusa is an Associate Professor at the University of Southern California's Viterbi School of Engineering and Rossier School of Education. She has expertise in engineering education, design pedagogy, faculty development, K-12 STEM education and assessment, measurement, and advanced research design.

## A Practical and Comprehensive Approach of Assessing ABET Outcome Achievement in Computer Science and Computer Engineering

introduction

Every serious organization has a vision or mission statement. Schools of Engineering are not exempt from this charge. Though missions and visions are easy to write, their achievement is often difficult to assess. The Accreditation Board of Engineering and Technology (ABET) has crystallized this issue among university undergraduate engineering programs attempting to gain accreditation. ABET requires that an undergraduate program state its educational outcomes and demonstrate how it measures outcome achievement. Many programs, including ours in computer science, have in the past relied almost exclusively on course specific student perception surveys and other indirect methods of student assessment. These "perceptions" have been largely discredited as biased and subjective. In recent years, ABET has challenged the academic community to utilize assessment methodologies based on direct, measurable data. Our response, the subject of this ASEE paper, proposes a methodology that requires professors to state their individual course outcomes and map them to the ABET program outcomes, and produce for each exam or assignment three important components comprising: (1) the source document, (2) a mapping of this exam or assignment to the class outcomes, and (3) the results. Our 2-level outcome and assessment metrics mapping supports precisely the kind of outcome-achievement analysis that ABET desires. Accordingly, and perhaps because this methodology is easy to explain and interpret, we have achieved 100% compliance with our undergraduate teaching faculty.

ABET requires many criteria be met for accreditation. Criterion 2 lists Program Educational Objectives that graduates are expected to fulfill during their professional careers. For example in computer science, Criterion 2.3 asks whether “Graduates have followed a career path for which they have been trained either through suitable employment or graduate studies.” Alumni surveys are the standard way to get answers. However, contacting the companies and asking about USC engineering students is difficult; most consider privacy issues enough to squash this idea. The alumni themselves are hard to find. ABET seems to sense the difficulty and our evaluators were satisfied with our limited survey results.

Criterion 3’s program outcomes are a different matter. These are the abilities that students are supposed to have on graduation from your program. For example, the a-k outcomes that ABET suggests for a computer science program are:

a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;
b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
d) An ability to function effectively on teams to accomplish a common goal;
e) An understanding of professional, ethical, legal, security, and social issues and responsibilities;
f) An ability to communicate effectively with a range of audiences;
g) An ability to analyze the local and global impact of computing on individuals, organizations and society;
h) Recognition of the need for, and an ability to engage in, continuing professional development;
i) An ability to use current techniques, skills, and tools necessary for computing practices.
j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;
k) An ability to apply design and development principles in the construction of software systems of varying complexity.

These are abstract, easy to understand, but not trivial to assess. Our goal was to produce results like those shown in Table 1 below. Each cell represents the percentage of achievement of a particular outcome by a particular class that semester; if a class does not address that particular outcome, the cell is blank. The last row is the “average” achievement for each objective. This paper describes how we produce that table.

**Table 1: Computer Science ABET a-k Program Outcome Achievement Percentages by Course: Fall 2008**

| Class | a | b | c | d | e | f | g | h | i | j | k |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 101 | 79.4 | | 74.5 | | | | | 75.5 | | 71.5 | 76.7 |
| 102 | 75.6 | | 74.1 | | | 74.6 | | | | 72.6 | |
| 200 | | | 79.8 | | | | | | 79.8 | | 78.2 |
| 201 | | 86.8 | 88.2 | 93.7 | 88.7 | 90.8 | 89.7 | | | 89.0 | 89.8 |
| 271 | 90.9 | | | | | | | | | | 90.9 |
| 303 | 68.2 | | 68.2 | | | 67.2 | | | 71.6 | 66.9 | 67.2 |
| 351 | 68.8 | | 68.8 | | 62.5 | | | | 62.5 | 65.7 | 65.4 |
| 377 | 72.9 | | 70.5 | | | | | | 74.1 | 71.1 | 69.0 |
| 380 | 100 | 100 | 100 | | | | | | | | 97.6 |
| 402 | | | | 79.4 | | | | | | | |
| 445 | 83.0 | 90.7 | 90.7 | 90.7 | 85.8 | | 93.0 | | 93.0 | 85.4 | 86.4 |
| 477 | | | 61.1 | 95 | | | 61.4 | | | 61.1 | 61.1 |
| 480 | 70.7 | | | | | | 82.4 | | 58.3 | 78.6 | 78.6 |
| 485 | 65.5 | 63.6 | 66.7 | | | 59.3 | | | | 62.6 | 90.2 |
| 486 | 90.4 | 73.2 | 90.7 | | | 73.2 | | 90.4 | 90.4 | 90.4 | |
| 499 int. | 69.2 | | 69.2 | | 86.5 | 86.5 | 86.5 | | 95.1 | 68.1 | 70.2 |
| Average | 79.0 | 87.8 | 78.1 | 85.2 | 78.3 | 73.0 | 79.8 | 81.4 | 78.8 | 75.5 | 79.8 |

review of the literature

We live in an era with unprecedented changes due to dramatic advances in technology on many fronts. The explosive growth in computing and communication has revolutionized the way we work and live. Increasingly, the engineering work force requires that teams work with global foci. There have been many national level studies about critical issues facing the nation and related the crises in engineering education[3,4]. With outsourcing of engineering jobs, there is a growing concern about the level of interest among young students choosing engineering field. While the number of engineering graduates per year has remained steady at approximately 70,000 in the United States, in the past decade the number of engineering graduates per year from China and India has grown at a significant rate. With the world becoming "flat" due to globalization[2], increasingly, jobs requiring basic technical skills are moving beyond U.S. borders by companies in efforts to reduce costs. Engineering graduates from the United States must bring added value and higher-level skills including innovation, a problem solving approach, and leadership to garner higher salary jobs in U.S. companies[3]. The call from various technical reports on engineering education demands that U.S. higher education institutions produce this kind of engineers. Accordingly, there is an urgent need for reforming and enhancing engineering education to address these needs. This reform effort is best served through a merging of engineering education with best practices in educational psychology. The Accreditation Board of Engineering and Technology (ABET) is on board and fully supportive of the reform needs in engineering and computer science education. Accordingly, ABET now requires documentation of objectives and outcomes and associated direct assessment of student learning in undergraduate programs. As such, we have designed a research-based system to meet these challenges.

design and instrumentation

what we did before

Until this last ABET review, our primary assessment method was as follows. First we created a mapping of classes to the a-k. An entry was either blank (no correlation in this class to the outcome) or S (strong), M (medium), W (weak) correlation. **Table 2** shows the beginning of that table.

**Table 2: Classes mapped to ABET program outcomes a-k**

| | a | b | c | d | e | f | g | h | i | j | k |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **csci 101** | **S** | | **S** | | **M** | | | **S** | | **S** | **S** |
| **csci 102** | **S** | | **S** | | | **S** | **M** | | | **S** | |
| **…** | | | | | | | | | | | |

Next, we made sure each of the a-k was "covered" at least twice. Then, end-of-semester class surveys were given. Each professor lists course outcomes for the class and then asks students to rate their achievement. For example:

**End-of-year csci201 Course Assessment Spring 2008**
Instructor: David Wilczynski

Attribute of CSCI 201 How much did the course help develop the attribute?

| | Not at all | | | | A great deal | Not Applicable |
|---|---|---|---|---|---|---|
| (i) The ability to understand software engineering in terms of requirements, design, and implementation | 1 | 2 | 3 | 4 | 5 | NA |
| (ii) An understanding of how to use interaction diagrams to help define requirements. | 1 | 2 | 3 | 4 | 5 | NA |
| (iii) The ability to produce a software design based on requirements. | 1 | 2 | 3 | 4 | 5 | NA |

and so forth. A class could have as many attributes as expressed by the professor; for simplicity, only three of the thirteen attributes for the example CSCI201 class are shown. We did little with those surveys results. We simply did not know how they relate to the a-k. Some new mappings were needed.

the new methodology

Instead of mapping classes to the a-k, we now map the above class outcomes to the a-k. **Table 3** below shows such a mapping for the csci201 class. An entry is either blank (no correlation in this outcome) or S (strong), M (medium), W (weak).

**Table 3 - Class Outcome to Program Outcome Mapping for csci201 [Matrix 3 in Computation below]**

| | Class outcomes / Program Outcomes | a | b | c | d | e | f | g | h | i | j | k |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **i.** | The ability to understand software engineering in terms of requirements, design, and implementation. | M | | S | | | | S | | W | | S |
| **ii.** | An understanding of how to use interaction diagrams to help define requirements. | M | | S | | | | W | | | S | S |
| **iii.** | The ability to produce a software design based on requirements. | M | | S | | | | W | | | S | S |

This mapping is done once per course per semester. Every assessed class produces one. The mappings tend to be stable across semesters, but they can be reused or changed as each professor of the course sees fit. It takes about one hour for a professor to produce this mapping. Now the challenge is to assess the i, ii, iii, etc. without using surveys.

We do it using the tests and assignments of the class. All we need is one more mapping and some fine grained data as follows: for each assignment, project, or test the instructor must provide a triple:

1. A mapping from the item to the course outcomes. For a test it would map each question; for an assignment or project it would map the grading rubric.
2. The data showing how each student did on the individual questions (or rubric).
3. The actual exam or assignment (simply for documentation purposes).

Here is an example of the mapping for an exam that had four questions. Question 1 has an x in column ii, which, looking in Table 3, means it was about “An understanding of how to use interaction diagrams to help define requirements.” Some questions can and typically do have multiple x’s, meaning it addresses several of the course outcomes. Every item must have at least one x, otherwise why is the instructor asking that question?

**Table 4 - An Exam Mapping [Matrix 2 in Computation Below]**

| *Class: CSCI 201 Item: Midterm 1 Date: 10/1/08* | | | |
|---|---|---|---|
| **Class Outcomes to right. Question numbers down.** | **i** | **ii** | **iii** |
| **1** | | x | |
| **2** | x | | |
| **3** | x | | |
| **4** | | | x |

The associated test result data is shown in Table 5. This is the fine-grained data we collect. For our purposes we need to record all the scores rather than just the total. This is the most onerous part of the process; however the data entry is usually done by a teaching assistant or grader. We do not store the names or id's of the students. [The Total column is actually unused by us.]

**Table 5 - The Exam Data [Matrix 1 in Computation Below]**

| **Question** | **1** | **2** | **3** | **4** | **Total** |
|---|---|---|---|---|---|
| **points possible** | 15 | 20 | 30 | 35 | 100 |
| **Student 1** | 9 | 1 | 20 | 25 | 55 |
| **Student 2** | 14 | 7 | 24 | 30 | 75 |
| **…** | | | | | |

the intuition behind the math

Now it's simply a matter of some matrix algebra and aggregating the results to produce Table 1. We have decided (arbitrarily) to only use the S (strong) associations in Table 3. Before showing the actual math, we will first discuss the intuition behind the method.

We are going to multiply the matrix part of Table 5 (call it Matrix 1) times the matrix part of Table 4 (Matrix 2) times the matrix part of Table 3 (Matrix 3). The x's and S's will be replaced by 1's.

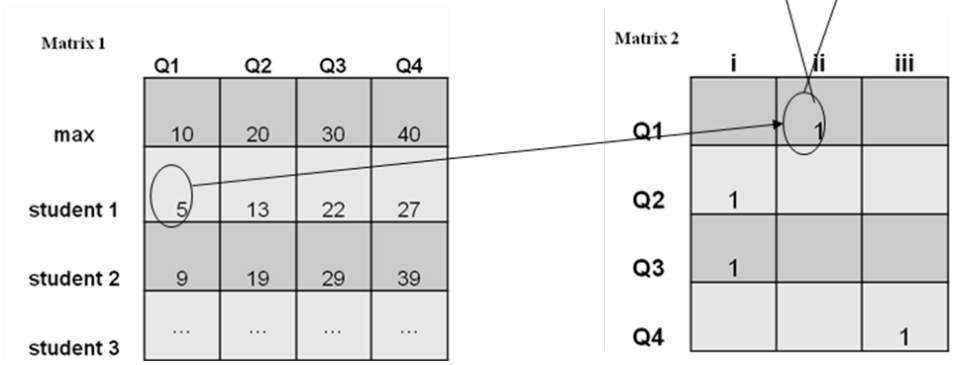
When row 2 of Matrix 1 multiplies column ii of Matrix 2, the result is:

5*1 + 13*0 +22*0 +27*0 = 5

This represents the test's contribution to class outcome ii by student 1. We can see that only question 1 affects outcome ii—Matrix 2 only has a 1 in the first location of column ii. Then when we multiply the result with Matrix 3, we can see that the value of Q1 gives evidence to program outcome c and k (since we are only mapping the S outcomes). When we multiply the **max** row of Matrix one to Matrix 2 and Matrix 3, we get the maximum evidence given by each question. So, in this example, student 1's 5 on Q1 gives 5/10 for his contribution. After this, we simply aggregate over all the students, and then all the exams and projects.

Matrix 3

| | csci201 Class outcomes | a | b | c | … | k |
|---|---|---|---|---|---|---|
| i. | The ability to understand the software engineering in terms of requirements… | m | | s | … | s |
| ii. | An understanding of how to use interaction diagrams to help define requirements | m | | s | … | s |
| iii. | The ability to produce a software design based on requirements. | m | | s | … | s |



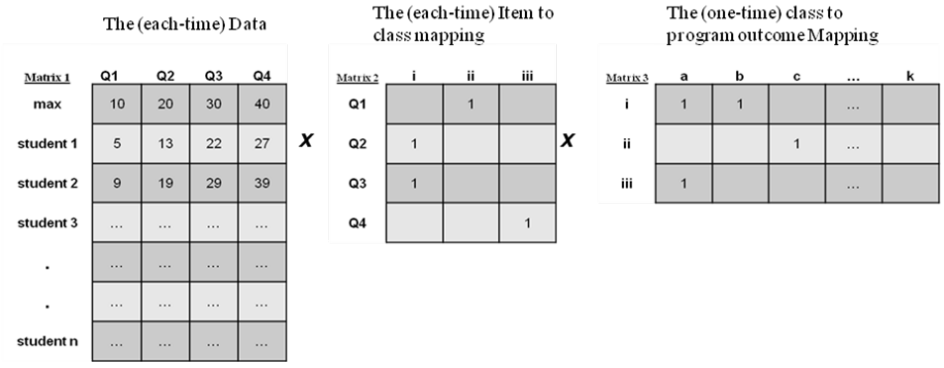

the math

Now we will look at how a row of Table 1 is computed. In the example below, we will be computing results for csci201.

Computer Science Outcome Assessment Fall, 2009

| | a | b | c | … | k |
|---|---|---|---|---|---|
| csci101 | .86 | NA | .75 | … | .80 |
| csci201 | To be computed | To be computed | To be computed | … | NA |
| … | | | | … | |
| … | | | | … | |

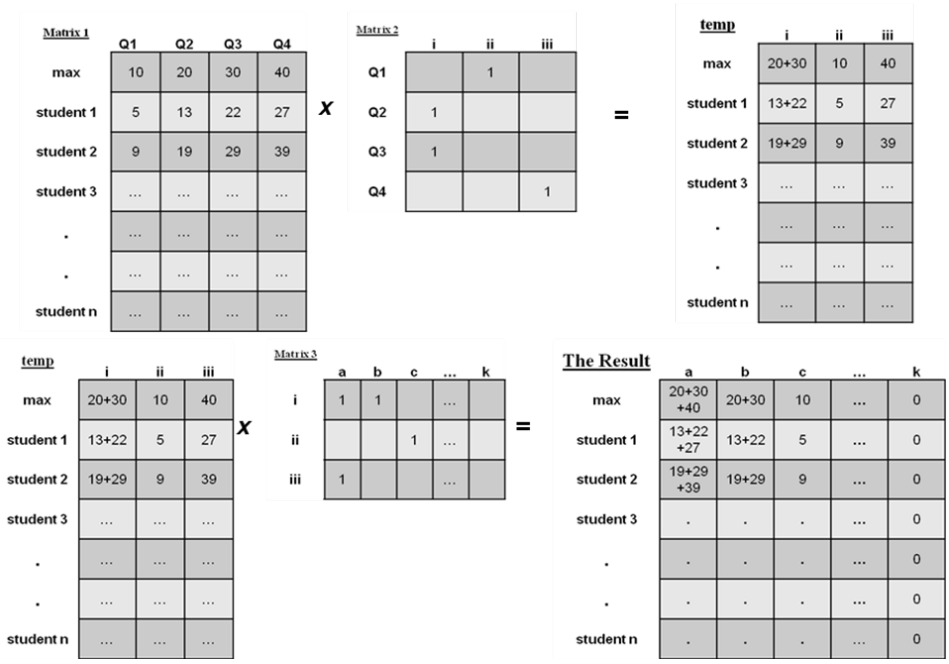
Matrix1 x Matrix2 x Matrix3 is this item's contribution to the Assessment Matrix




Remember that 1's replaced the S(trong) entries in Matrix 3. We could easily incorporated the M(edium) and W(eak) mappings but decided somewhat arbitrarily not to.

The matrix multiply is shown below in two steps. First the result of Matrix 1 * Matrix 2 is shown in temp. Then temp * Matrix 3 is the overall result of the matrix operation for this exam.

The Result = Matrix 1 * Matrix 2 * Matrix 3




Thus, The Result Matrix shows how each student contributed to the a-k with the row labeled max giving the total possible. So, for example, student 1's contribution to outcome b is: (13+22) / (20+30) = .7

Notice that all the entries in column k are zero. This could have happened for two reasons: (1) this exam had no questions that mapped onto a class outcome that mapped onto program outcome k; or (2) The class has no outcome that maps onto program outcome k. Both cases are ordinary and likely.

Now we aggregate over all the students to produce an *item* per a-k outcome. An *item* has three slots: integer TotalPossible, integer TotalPoints, and float weight. The calculation is as follows, where n is the number of students who took the exam:

```
For column = a to k {
   item.TotalPossible =  Matrix4[ max , column] * n ;
                         n
   item.TotalPoints   =  ∑  Matrix4[ i, column];
                         i=1
   item.weight = weight;    //provided with the item by professor
   csci201.column.items.add(item);  // append item to the list of items
}
```

| | a | b | c | … | k |
|---|---|---|---|---|---|
| csci101 | | | | … | |
| csci201 | { (TotalPossible=90*n; TotalPoints=62+87+…; weight = .25) } | { (Total Possible=50*n; Total Points=35+48+…; weight = .25) } | { (Total Possible=10*n; Total Points=5+9+…; weight = .25) } | … | 0 |
| … | | | | | |
| … | | | | | |

The weight is how much this item counts in the overall grade given by the professor. We store the list this way because it is likely that the weighting factors may not come until the end of the semester.

Once all the exams and assignments have been stored, we are ready to compute the final averages for the assessment matrix. Before showing the math, please note the following about the assessment table shown below:

- The example below shows that csci201 had two items (tests, assignments, or projects).
- The first was worth 25% of their grade, the second was worth 75%.
- The sum of the weights need not add to 100%. The class might have something that counts in their grade that is not being assessed, for example, attendance.

Now, it is simply a question of computing the average for each cell. The math is shown just below:

| | a | b | c | … | k |
|---|---|---|---|---|---|
| csci101 | | | | … | |
| csci201 | {<br>(TotalPossible=90*n;<br>TotalPoints=62+87+…;<br>weight = .25)<br>(TotalPossible=70*m;<br>TotalPoints=60+65+…;<br>weight = .75)<br>} | {<br>(Total Possible=50*n;<br>Total Points=35+48+…;<br>weight = .25)<br>(TotalPossible=80*m;<br>TotalPoints=62+70+…;<br>weight = .75)<br>} | {<br>(Total Possible=10*n;<br>Total Points=5+9+…;<br>weight = .25)<br>(TotalPossible=90*m;<br>TotalPoints=88+83+…;<br>weight = .75)<br>} | … | 0 |
| … | | | | | |
| … | | | | | |

```
For column = a to k {
  csci201.column.average = 0;
  for item in csci102.column.items
       csci201.column.average +=
                  item.weight * item.TotalPoints / item.TotalPossible;
}
```

the program's average—computing the last row of Table 1

The above computation produces everything in Table 1 except for the last row, the overall program average for each of the program outcomes. For that result, we just took the average of the non-zero elements in the column. Considering that one class may have a result in that column based on one data point while another class has twenty leads us to believe that a simple average may not be adequate. On the other hand that single data point may represent the entire output of a class. More study is required here.

normalization and scaling

In the discussion above, we have ignored normalizing results from different professors. That is, suppose one professor has classic scores, like [90-100] is A, [80-90) is B, etc. Another professor's tests may be scored differently. For example, the professor might say

[80-100] is an A, [60-80) is a B, etc. If we are to unify the items of both professors, we need to normalize the results. This is very easy since it is just interpolation.

Our normalization is somewhat complicated because we keep our results as lists of the results of each question. Before showing how that works, here's how normalization would work if a student score was simply one total.

| | Normalized Range | Prof. 1 Submitted Range |
|---|---|---|
| **A** | **[90, 100]** | **[80, 100]** |
| **B** | **[80, 90)** | **[60, 80)** |
| **C** | **[70, 80)** | **[30, 60)** |
| **D** | **[60, 70)** | **[10, 30)** |
| **F** | **[0, 60)** | **[0, 10)** |

Let S be a submitted score to be normalized.
Let g be the grade of S (using the Prof's ranges).
Let $N_S$ be the normalized score for S to be computed.

Let $N_{g\text{-}top}$ be the top of the normalized g-range.
Let $N_{g\text{-}bot}$ be the bottom of the normalized g-range.
Let $S_{g\text{-}top}$ be the top of the submitted g-range.
Let $S_{g\text{-}bot}$ be the bottom of the submitted g-range.
Then $$N_S = N_{g\text{-}bot} + [(N_{g\text{-}top} - N_{g\text{-}bot}) / (S_{g\text{-}top} - S_{g\text{-}bot})] * (S - S_{g\text{-}bot})$$

Note that everything after the "+" sign is just interpolation**.**
For example, if Prof. 1 submits a student score of 86 it is normalized as follows:

$$N_{86} = 90 + [(100\text{-}90)/(100\text{-}80)] * (86\text{-}80) = 90 + (10/20) * 6 = 93$$

However, as we said above, for each exam a student score is a list like [5,13,22,27], where each entry is the individual scores on the exam's questions. Each member might be scaled independently and could have a separate range matrix. That's too much! We'll just use the one scaling matrix as follows:

| | **Normalized Range** | **Prof. 1 Submitted Range** |
|---|---|---|
| **A** | [90, 100] | [80, 100] |
| **B** | [80, 90) | [60, 80) |
| **C** | [70, 80) | [30, 60) |
| **D** | [60, 70) | [10, 30) |
| **F** | [0, 60) | [0, 10) |

Let S[ ] be the submitted vector to be normalized.
Let $S_{Max}$ [ ] be the max vector for this item.
Let g be the grade of S[i] (using the Prof's ranges).
Let $N_S$[i] be the normalized score for S[i] to be computed.

Then $N_S[i] = S_{Max}[i] * Scale(S[i] / S_{Max}[i]*100)$ where Scale(S) is the scaling function above.

E.g., if Prof. 1 submits S[ ] = [5,13,22,27] whose max is $S_{Max}$ [ ] = [10,20,30,40], normalization for the first and third term is:

$N_S[1] = S_{Max}[1] * Scale(5/10*100) /100$ //5 out of 10 is 50% which is 2/3 up on the C's
$= 10 * (N_{g\text{-}bot} + [(N_{g\text{-}top} - N_{g\text{-}bot}) / (S_{g\text{-}top} - S_{g\text{-}bot})] * (50 - S_{g\text{-}bot})) / 100$
$= 10 * (70 + 10/30 * (50 - 30)) / 100$
$= 10 * (70 + 1/3 * 20) / 100$
$\sim 7.667$

$N_S$ [3] = $S_{Max}$ [3] * Scale (22/30*100) /100 //22 out of 30 is 73.33% which is over 1/2 up the B's
= 30 * ($N_{g\text{-}bot}$ + [($N_{g\text{-}top}$ – $N_{g\text{-}bot}$ ) / ($S_{g\text{-}top}$ – $S_{g\text{-}bot}$ )] * (S – $S_{g\text{-}bot}$ )) / 100
= 30 * (80 + 10/20 * (73.33 – 60)) /100
= 30 * (80 + ½ * 13.33) / 100 ~ 26

This produces the scaled list for each student's results on this exam. It is possible that the professor could give us a scaling matrix for each test or assignment. Why not? Each sub-item could have its own scaling factor. In practice, we get a single scaling matrix at the end of the semester, which is a fair approximation considering all the averaging we are doing. Wilczynski, in fact, gives scores which scale every question on every exam to the standard normalized ranges so students know exactly how they are doing at all times. Some instructors do all their scaling at the end with one scaling matrix. Either method is acceptable.

analysis

Methodologically, to achieve our goals of direct assessment of student learning in accordance with ABET criteria and to articulate our program mission in alignment with ABET's charge, we have employed both descriptive statistics and a normalization technique of student assessment data analyses. An information processing approach to student achievement has served as our theoretical frame for which we developed our assessment metrics. This theoretical frame provides the most robust means for which to engage in direct assessment of student achievement and to map achievement to program outcomes. Specifically, once mapping between course objectives, ABET outcomes and student assessment metrics was completed, resulting student scores by outcome were summed and then averaged (our normalization technique). Importantly, these scores were weighted up front by the professors as a function of the amount of weight they placed for each assignment in the course total. The averaged outcome achievement scores were then mapped back course-by course. Accordingly, we created a complete mapping of ABET outcome achievement by semester that is now being used for curriculum reform and program improvement in our computer science and computer engineering programs.

This comprehensive student achievement mapping process is mutually understandable by diverse faculty and somewhat labor neutral for academic departments once the initial structures for mapping, data collection, and analyses are established and support is provided. It can be adopted and adapted to fit with diverse engineering and science academic departments as a means of analyzing diverse data sets and mapping student achievement to program outcomes.

results

After the ABET visit in fall 2009 during which our department passed its accreditation, the Computer Science Curriculum Committee made some recommendations to the Chair, which have been approved:

1. A faculty member will be responsible for the “ABET process” described in this paper.
2. A student will be hired each semester to assist that faculty member. [This student is being paid $15/hour for 10 hours per week out of the department budget. The current student is developing a Java application to do the work described in this paper.]
3. Only required classes will be part of this “ABET process.”
4. The Curriculum Committee will meet on a regular basis and will:

    - *Analyze the data to see if a class is underperforming*. For this first set of data we used a 75% pass rate (after checking for mistakes). We expect the pass rate to be 80% for the 2009/2010 academic year. This is the primary reason for collecting the data.
    - *Review all class-outcome-to-program-outcome mappings to eliminate faculty bias or plain mistakes*. Over the course of the fall/spring semesters we will look at all the mapping. In cases of discrepancies we will invite the faculty member to our meeting and have them justify their mapping and measurement techniques. A faculty member teaching an ABET course for the first time can make a new class outcome to program outcome mapping. In that case, that mapping is reviewed immediately.
    - *Pick some representative samples of tests and assignments and review them vis a vis the outcome mappings*. For example, we were surprised to see a result in CSCI 102 for outcome f: An ability to communicate effectively with a range of audiences. It turns out the instructor has mapped writing good code to the communication outcome. Do we agree with that? It’s not clear. But after a thorough review we can point to "best measurement practices" and spread it throughout our faculty.

By nurturing the data collection process this way, the CS/CECS Curriculum Committee will make it a sustainable and invaluable part of our process to improve our program.

conclusions, discussions and contributions to the field

The process of comprehensively assessing ABET outcomes with students is a difficult one, one that requires department-wide buy in for systems to be effective. We have designed an inclusive system that faculty are responsible for and that is currently being fully implemented in all of our ABET accredited bachelor’s degree courses in computer science. At this point in our ABET assessment process aligned with outcomes a-k, we have simply employed this new cross-course alignment and score normalization with a-k outcome alignment. As a consequence, we now have direct assessment of each ABET outcome rather than simply a measurement of students’ perception of whether they have been exposed to each outcome. We have yet to assess whether our process is working from the standpoint of faculty who teach the computer sciences courses. Accordingly, we plan to convene focus groups with faculty in the coming semesters to determine if the new system is effective as a means of offering continuous improvement to faculty courses in terms of improving curriculum and pedagogy based on assessment results. The focus group data will be used to continuously improve the process, so that we have a feedback loop with faculty about the successes and challenges for the process. Our intent is to document

the changes that faculty make in their courses as a function of the assessment data that they receive from their students. This will serve as a means of ensuring that course adjustments are guided by student assessment data which is akin to data driven decision making[1]. This process and its comprehensiveness contributes greatly to the field because it can be mapped onto other bachelor's degree programs in CS and other engineering majors. Additionally, due to the fact that it is highly comprehensive, it can inform assessment practices in graduate education, particular at the master's level where courses are highly aligned with the practice needs of engineering industries.

Perhaps the best thing about our process is that ABET accepted it during our 2009 accreditation cycle. That is the best assessment of our assessment process.

references


1. Armstrong , J. & Anthes, K. (2004) How Data Can Help: Putting Information to Work to Raise Student Achievement. American School Board Journal. 24 (2). 209-214.
2. Duderstadt, J. J. (2008) Engineering For a Changing World: A Roadmap to the Future of Engineering Practice, Research and Education. The Millennium Project. University of Michigan.
3. National Academy of Engineering, (NAE, 2004), *The Engineer of 2020*, National Academy of Engineering, The National Academy Press, Washington DC.
4. National Academy of Engineering, (NAE, 2005), *Educating the Engineer of 2020*. National Academy of Engineering, The National Academy Press, Washington DC.
5. Prince, M.J., and Felder, R. M. (2006). Inductive teaching and learning methods: definitions, Comparisons, and research bases. *Journal of Engineering Education.* Vol 101. 4 123-137.
6. Shuman, L. J., Besterfield-Sacre, M. and J. McGourty, (2005) "The ABET Professionals Skills – Can they be taught? Can they be assessed?" *Journal of Engineering Education*, Vol. 94, No. 1, pp. 41-56.