A Multidisciplinary Capstone Teaching Model - An Integrated, Multilevel, Intradisciplinary Engineering Design Approach

Lt. Col. Anthony S. Ruocco and MAJ Roger S. Dixon
Department of Electrical Engineering and Computer Science
United States Military Academy
West Point, NY 10996

Abstract - This paper presents the design of a curricular program that meets the educational needs of an interdisciplinary student cohort (e.g. where many are not majoring in any engineering discipline) through courses with intradisciplinary involvement between different stages in the instructional sequence. It proposes a curricular paradigm which allows students to work in groups on a single, large, real-world problem over multiple terms. Students from senior level courses act as the group leaders/mentors for student groups from lower level courses. This teaching method will provide the student with an engineering design experience more consistent with his/her eventual real-world experience than more traditional curricular paradigms have allowed.

Introduction

The purpose of the United States Military Academy (USMA) at West Point is to provide the nation with leaders of character who serve the common defense. The Academy fulfills this requirement through training cadets physically, militarily, and by providing them with a broad undergraduate education. USMA offers the Bachelor of Science degree in 27 areas. The heart of all the programs is a broad core curriculum divided, almost equally, between liberal arts and the sciences. The core program provides the underlying foundation for each cadet, whatever major, to take a mandatory five-course engineering sequence. The remaining courses needed to complete a degree come from the cadet’s area of interest (major). Currently there are seven sequences covering Civil Engineering, Computer Science, Electrical Engineering, Environmental Engineering, Mechanical Engineering, Nuclear Engineering, and Systems Engineering. Each sequence culminates in a capstone engineer design project which enforces the engineering process of "analyze, design, build, and test." For many cadets, the sequence is imbedded in the chosen major. That is, those majoring in Electrical Engineering take the Electrical Engineering Sequence. Course scheduling does not differ for those taking just sequence courses (from now on called Sequencers) from those majoring in the field (Majors).

Thus, any given course is composed of an interdisciplinary mix of Sequencers and Majors. This provides a unique challenge to the instructor as he or she tries to teach to the varied levels of understanding and interest in the subject matter. We will focus on the field of computer science as we explain a model that capitalizes on this interdisciplinary composition of the student population. This model not only emphasizes the interdisciplinary nature of any given course, but also captures the experiences of previous students as we integrate those more experienced students into the teaching and learning process. The model therefore reflects an interdisciplinary and an intradisciplinary approach to teaching. The model, though described in terms of computer science, can be extended to other fields within USMA, and to other universities as well.

The current method:

The courses which make up the computer science sequence must meet the needs of the student majoring in computer science, and meet the needs of the CS Sequencers. The five courses chosen for the engineering sequence provide a solid background for officers entering today's high-tech Army. These courses are: Fundamentals in Computer Science, Information Systems, Database Systems, and the capstone courses Software Design I & II. The first two academic years are oriented on the core curriculum, so the sequence is designed to be accomplished in the final two years. Basically, the courses are taken in the order listed. The fundamental course in the first semester of the third academic year, the information systems and database courses the next semester, the first design course in the semester before last, then the second design course in the last academic semester. The first four courses follow a traditional teaching and evaluation model, that is, the use of many individual and/or group software development projects within each course. Course projects are targeted to reinforce the goals of that particular course, and typically culminate in a significant project encompassing a wide-range of course objectives. However, each project tends to be small, and has
a limited scope to allow the students to complete the project within a given time frame. The capstone courses tend to concentrate on larger projects. The final course in the sequence, Software Design II, is typically designed as a single semester project with cadets providing a functional product to a non-academic oriented user.

There are several shortfalls in this traditional approach. Each course, though linked by curricula considerations, tends to be viewed by cadets as an independent entity. That is, once a course is done, there is little follow-up. For example, while there is constant emphasis on writing maintainable code, cadets rarely, if ever, are tasked to update or re-engineer projects from previous courses. Even with the larger efforts of the capstone courses, projects don’t come close to preparing the student for the issues involved in developing a system of the size and complexity of most real-world systems an officer is likely to face within the military (or within industry).

Typically, a significant project not only requires a group effort, but that the individuals’ efforts be channeled. This is done by either formally, or informally, selecting a group leader. This leader must then face the challenges brought on by group dynamics. In many cases, the group leader may not be able to learn anything about the subject matter because he or she is so wrapped up in the direction the group members are going, an effort similar to trying to herd cats. In other words, the group leader gets a good, and valuable, experience in leadership but has missed the learning objectives the group project was targeting. Conversely, and often disregarded, the group members have not had the experience of dealing with group dynamics. Therefore both parties have learned something from the group effort but at the expense of something else.

The proposed model

We propose a variation of the above model to better introduce the students to the issues they will face with real world systems. We concentrate on a large-scale real-world problem such as the migration of a legacy system. Examples of suitable problem systems within the Army could be the Total Officer Personnel Management Information System (TOPMIS) or the Standard Army Management Information System (STAMIS).

In the Fundamentals course, groups concentrate only on the analysis and design issues involved with the selected problem. The problem is integrated into each of the follow-on courses in the sequence. Groups concentrate on the issues appropriate for that particular course (i.e. interfaces in Information Systems, back-end issues in Databases, and systems integration in Software Systems Design). However, while the large-scale project permeates through all courses, an instructor is still free to use small projects which emphasize specific course objectives. These small projects are independent of the larger problem.

Groups for them can be either the same or different from the large problem based on the discretion of the instructor. Since each sequence course has numerous projects anyway, the instructor uses the large-scale problem as the basis for the course’s "significant" project. The course instructor also incorporates individual efforts as necessary to meet course objectives. Under this model, the student can be exposed to realistically large systems issues, and can see how each of the topics identified in all the sequence courses builds on and reinforces the others. In addition, we propose a major difference in selection of groups’ leaders.

While most models allow either a group-elected leader, or an instructor directed leader, we take leaders from outside the course. Group leaders come from the senior level courses. In other words, the people leading each of the groups have already had the subject matter for that particular course.

Those designated as group leaders can concentrate on controlling group dynamics while group members learn the course objectives. Having already taken the course, the group leader is experienced. He or she can provide experienced guidance toward the goal. Since the group leader is not a peer of the group members, he or she is in a position to single out those not fully participating. The group leader can take these people aside and help them overcome the lack of participation in the group. Should group leader intervention not provide the necessary incentive, the group leader can contact the course professor. While different from providing a "firing" mechanism, the individual group members are aware that there is someone who is monitoring their individual progress as well as the group progress. Group members are aware of the group dynamics, but all can concentrate on the course/project objectives. Not only can they concentrate on objectives, but they are exposed to their group leader’s style and methodology. Thus, when they move on to more senior courses, they will have been exposed to a variety of group leader styles. They can then apply the techniques they felt worked best when they were group members on their own groups.

Model Implementation

The proposed model offers many advantages over traditional approaches. However, there are several concerns that must be considered. Obviously, the key is to have a large-scale problem with elements from each course in the sequence. There would actually need to be several problems being developed. The cadets in the fundamentals class begin the requirements phase of large-scale problem 1. Then, as seniors, they are still working on the maintenance and upgrading of large-scale problem 1. But, as seniors, they are also the group leaders for upcoming class.
Therefore, while working on large-scale problem 1, they are group leaders for cadets developing requirements for large-scale problem 2. Cadets then find themselves in different, concurrent roles. That is, while group members for aspects of problem 1, they are group leaders for aspects of problem 2. That means not only are cadets facing multiple problems, but so are the instructors. Instructors are evaluating group members on meeting course objectives in the group work, while simultaneously evaluating the group leaders independently of the problem.

Besides the added evaluation burden on course instructors, there is a large administrative burden placed on the course directors of the senior level design courses. The seniors in these courses must be given equal opportunities to be group leaders. However, there is a wide range of capabilities within the senior courses. The course director can take advantage of the differences between Majors and Sequencers by assigning Sequencers to groups working requirements and analysis (the fundamental course) and Majors to groups working database or interface issues. In this way, the experience of a broader range of computing classes taken by majors can be brought to bear on the courses dealing with implementation tradeoffs. At the same time, the Sequencers can bring to bear their experiences with non-computer courses as they deal with the interaction of "users" and "builders" so vital to successful requirement definition and analysis.

The scheduling of courses should not be a concern. All students must take the fundamentals course as the first course in the sequence. Therefore, each group devises its own set of requirements for the large-scale problem in this course. As students move through the sequence these requirements follow along as well. It is not necessary for the groups to be the same from course to course. Only the number of groups is constant. By changing groups, cadets are exposed to the idea that while large-scale problems extend over years, team composition changes frequently. Cadets also realize that each piece of the design process is important as they realize they may need to implement a problem based on someone else’s design. Allowing scheduling to take place independently of the group-work is a valid simulation of the change students will face outside an academically controlled arena. Group leaders are also scheduled into their senior-level courses. This means senior course instructors must adjust for student group leader commitments. One approach would be to consider this as a senior project, and account for it as part of the overall grading scheme.

**Applying the model to other disciplines**

So far, the model has been described based on its applications to computer science. But, it can be easily extended to the other engineering sequences at USMA. To verify this, we must show a reasonable mapping of the CS approach to the other engineering sequences. Each sequence follows the same basic pattern. There is a fundamentals-type course, followed by two courses oriented toward "breadth" of the discipline, and finally two courses oriented toward "depth" of the discipline.

Under the current model, the five-course sequence is already designated such that each course builds on the previous courses. This means a course cannot independently make radical changes that will effect all follow-on courses. Using an integrating problem however does not effect this. Instructors are free to make the same level of change in each of their courses, and radical changes still must be coordinated with all follow-on courses. Many of the senior design courses are dedicated to group work. In fact, many are actually a single project taking the entire semester. Our model does not change that. There is no reason a senior student cannot be working on a semester long group project, and still serve as a group leader for one of the junior courses. In fact, the multiple requirements better mirror the environment cadets graduate into than the traditional model of dedicating all efforts to a single endeavor. Again, the senior level instructors need to incorporate group leading time requirements into their overall course schedule.

**Conclusion**

The proposed model offers the ability to capture the synergism of group work at several levels. In addition, the threaded use of a single large-scale problem effectively ties together the five courses of the sequence. Students truly build on previous work. The current mentality of throwaway programs is averted. Using senior students as group leaders
provide the senior students the opportunity to mesh directing a group with meeting project deadlines. Using senior group leaders allows group members to concentrate on the group mission, and gain exposure to a variety of leadership styles. The senior students are not affected by peer-pressure within the group, hence are free to identify students who might otherwise ride-along with the group effort. Junior members are exposed to a variety of leadership techniques, yet still can concentrate on course objectives. Using Sequencers as group leaders captures the synergism brought about by different perspectives on a problem. While Majors focus on the nuts and bolts of building, the Sequencers bring in the perspective of non technical users, and user oriented concerns. Having a variety of leaders with different perspectives, coupled with an interdisciplinary mix of majors and Sequencers, is a much better reflection of the society into which we expect, and demand, our graduates to function.