The Studio Approach to Liberal Arts Computer Science[1]

James S. Bohy
Mount Mercy College
Department of Math and Computer Science
Cedar Rapids, IA 52402
jbohy@mtmercy.edu

Steve Strong
Mount Mercy College
Department of Math and Computer Science
Cedar Rapids, IA 52402
sstrong@mtmercy.edu

Abstract

In 2005, Mount Mercy College received a federal U.S. Department of Education Title III Strengthening Institutions Grant, primarily targeted at program improvement in the disciplines of nursing and computer science. The goal for the computer science program included increasing enrollment through the integration of technology into the curriculum. In the first three years of the Grant, renovation was completed and equipment was purchased for a dedicated computer science laboratory and faculty developed and piloted several new courses. In fall 2008, work began on a complete overhaul of the curriculum.

In response to current research, the demands of modern students for more interesting and innovative teaching methods, and the goals of the Title III Grant, the curriculum model chosen is interdisciplinary. All courses will have significant laboratory and project components. The projects will be assembled into portfolios that will serve the dual purpose of providing assessment data for the department as well as aiding individual students in career pursuits or graduate school preparation. Our discussion will focus on the various challenges encountered in developing the model, complete descriptions of the sample programs, and the framework that will be used to assess the programs (an important part of the Title III mandate).

[1] Many thanks to Marilyn Lefebure, the college's grant coordinator, for her contributions to this work.

Introduction

Mount Mercy College is a small, Catholic college in Cedar Rapids, Iowa. Known for its Education, Nursing and Business programs, it has had a Computer Science program for at least 20 years. By the summer of 2008, the Computer Science program at Mount Mercy College was at a crucial point in its evolution. Enrollments of 100 students had slumped to less than 40 and the college was actively considering dropping the program.

In an effort to rebuild the program and improve its curriculum and instruction, a US Department of Education Title III Strengthening Programs Grant proposal was submitted, in conjunction with the college's nursing department. The Grant was awarded in the fall of 2005. With the injection of much-needed funds, a new CS lab was constructed including state-of-the-art networking and server support.

Our primary goals were to develop a program that was unique enough to be attractive to a wide range of students, responsive to current needs, and flexible enough to respond to the changing needs of the workforce.

In our interdisciplinary model, all students will complete:

- a common core (6 courses)
- an *area of specialization*, developed in conjunction with the student's faculty advisor and faculty from other academic departments, which will include
    - a minimum of two computer science electives
    - a minimum of four courses outside of computer science
- a senior project related to their area of specialization

There were four main factors which influenced the development of the model. First, the department has limited human and financial resources; the model must make maximum use of these resources without putting undue strain on them. Second, the department serves a significant number of non-traditional students, most of whom receive tuition assistance from local employers in technology fields; the curriculum needed to appeal to these employers and have a generally broad appeal. Third, we felt obligated to demonstrate our effectiveness to our students and the community. Finally, the college is located within 75 miles of at least a dozen other post-secondary institutions. Thus, it was important that our program be distinctive.

History

As previously stated, Mount Mercy College has had a computer science major for more than 20 years. The program has maintained excellent relationships with local employers, including firms focused on more scientific/engineering aspects of computing (e.g., Rockwell-Collins and Intermec) as well as those focused on business and information technology development and applications (e.g., Aegon and Quaker Oats). The department has traditionally been staffed with two full-time computer science faculty,

and two full-time mathematics faculty. There has been some amount of turnover in the computer science faculty during that time.

Before making any meaningful changes, we needed to understand the current state of the Computer Science Program at Mount Mercy College. By the start of the fall term in 2008 much, if not all of the infrastructure necessary for the rebuilding of the CS program was in place. A linux server as well as a lab of 25 linux clients with state of the art programming and instructional software was in use. The lab is staffed and maintained by a part time IT staff person who knows the special requirements of CS instructional labs and who works with us to make sure the lab is always up and running. We have found this resource to be invaluable. Most computer science departments are not so fortunate. From a facilities point of view we were ready!

Still, it was very clear that there were several significant issues that needed to be addressed. These issues fell into roughly four categories:

- enrollment was low, going down and more than 60% of those enrolled were non-traditional students
- curriculum structure and design was confused, difficult to staff and not pedagogically questionable
- assessment of student progress and of instructional methods was almost non-existent
- documentation and formalization of program processes were limited and often not followed

In this paper we will focus only on the second item in the list, but it was clear from the beginning that all of these issues were inter-related and needed to be addressed as a whole.

Assessment of Current Curriculum

Our most significant challenges are related to the design and administration of our curriculum itself. The current catalog shows a large number of courses in support of three different programs of study; these things were at odds with the limitations in resources. The program required students to take courses in no fewer than four programming languages, but offered little depth. Finally, there was not a discernable curriculum development process in place.

Numbers Problems Galore

The 2008-2009 Mount Mercy College catalog [5] lists a total of 27 courses in computer science. While this number does include internships, it is a tall order for a department of two faculty members. Exploration of the offerings at other small private colleges in Iowa with similar teaching loads to those expected at Mount Mercy showed offerings of 29

courses with 3 full-time faculty members [8], 18 courses with 2 full-time faculty members [1], 30 courses with 4 full-time faculty members, 13 courses with 1 full-time faculty member [4], and 24 courses with 2 full-time faculty members[7]. While 27 classes is not horribly out of line, there were other problems with the structure of the curriculum that lead us to conclude that changes were imperative.

The same catalog shows the department supporting three programs of study: computer science, computer information systems (CIS), and management information systems (in conjunction with the business department). Typically, CIS programs are an interdisciplinary mix between computer science and business, similar to MIS, but heavier on the computer science content than the business content. The CIS program at Mount Mercy was clearly targeted at giving graduates the knowledge to pass CISCO certification exams, with 4 required courses in networking. The equipment for this program had become outdated, enrollments were dwindling, and the faculty who had designed it were long gone.

Finally, the institution had seen dwindling enrollments similar to the national trend. There are currently 41 students on campus with a declared major of computer science. Nearly 60 percent of those are non-traditional, part-time students. In the 2008-2009 academic year, 6 courses had to be cancelled due to low enrollment, and at least two have been allowed to run with less than minimum enrollment due to student need.

Prerequisite issues

In the current catalog, there are four courses with “programming in” in the title, including C, C++, Java, and Ada. The catalog descriptions of these courses clearly indicate that they are to be offered at an introductory level, with a clear perquisite shown on only one of them One student remarked that he'd taken the same course three times in three different languages and still couldn't do object oriented programming very well.

Most of the upper-level courses have mismatched or inappropriate prerequisites. The result is that the higher level courses often degenerate into review classes and students (especially gifted ones) are left without learning much. In addition to leading to some curricular redesign, we realized that we would need to shore up and formalize our advising process and hold to our standards for entrance into each upper level course.

Little or no on-going processes for instructional design and evaluation

Very little information existed in the department regarding progress. As part of the Title III Grant, several *pilot* courses were developed, with the idea being that these courses would become part of the new program. However, it was apparent that little work had been done in the way of formal course development and assessment. Documentation of development and assessment of courses in the existing curriculum was non-existent.

Moving Forward

Clearly, there were some significant challenges to face. As a result, several strategies were employed throughout the redesign of the Mount Mercy College CS curriculum to address these challenges and to accomplish as much as possible for our students, abiding by the constraints on resources, the highly non-traditional make-up of our student body, and the desire to offer a program that is unique and flexible. Throughout the redesign process we have looked for every opportunity to:

- *Offer students the opportunity to personalize their education* – we are offering a "Studio" approach to our Computer Science major. Students will be able to tailor their graduation plan to meet their individual needs. In this way, we can offer a more individualized program that is more diverse because these individual plans will require the student to take courses outside their major.

- *Design and implement more "hands on" teaching activities* – "Hands on" learning is more effective and more fun. Therefore, we will look for as many opportunities as possible to create active teaching environments and reduce lecture and demonstration. Special effort will be placed on the introductory classes to ensure that there is plenty of time for students to do lots of programming projects that are interesting and that demonstrate real-world uses of computing.

- *Increase the number of semester hours per course to four* – All programming classes will be four credit classes: three hours of lecture and two hours of structured lab activities. This design will allow us to use more "hands-on" activities which is at the core of our educational philosophy. From a completely pragmatic view, we can also teach the same number semester hours with fewer courses. This allows us to get good instruction to our students with a very small faculty. To ensure that all of the objectives still get taught, we will shift objectives from those courses we plan to eliminate to the new and revised courses we are developing.

- *Add two hours of lab instruction to almost every course* – We believe that Computer Science is only taught well with the use of structured lab sessions that require students to design and implement real programs. In addition, offering lab instruction in each course allows us to include more content as well as more time for guided practice and "hands on" learning activities.

- *Reduce the number of courses* – we have dropped fifteen courses from our curriculum. Semester hours taught will be down to 54 from 73. The dropped courses were highly specialized, often had very low enrollments and a few of them taught the same concepts as other courses using different programming languages. We also offer a rotation of topics courses to add flexibility.

- *Ensure that all programs have good enrollment and potential for growth* – we have dropped the Computer Information Services major. There were only five students currently in the major and this curriculum is available elsewhere for less cost and including preparation for industry standard certifications. Further, the equipment and maintenance costs that are associated with these courses are very high.

- *In the same vein, offer programs that are current and topical* – by streamlining the curriculum and adding a set of topics courses that can be taught by experts in the various fields, the curriculum should stay relatively fresh.

- *Design, Implement and Adhere to a standard set of Policies and Procedures*. – We will create flexible policies that can be easily followed in a consistent way.

All of this leads to what we are calling the *studio approach*.

The Studio Approach

The idea of a studio approach is prevalent in the fine arts, architecture, and various other disciplines. In music, the *studio* is where a student and instructor meet on a weekly basis to have focused instruction on the instrument or voice. The instructor provides the student with various activities to complete to develop their skill – some are fundamentals (e.g., scales), where other activities are related to the individual student's particular gifts. In art and architecture, the *studio* is a place where students go for focused practice, typically on real-world problems [6].

Our application of the term studio is a bit different from these. At the outset, the program looks fairly traditional. Students pursuing a degree in computer science from Mount Mercy will complete a core of six common courses in computer science and mathematics, to include the CS1/CS2 sequence, computer organization, systems programming, information ethics, and discrete mathematics.

At some point during their first two years, however, each student will be encouraged to work with their faculty advisor as well as faculty from other departments to develop an *area of specialization*. This specialization will include some number of electives in computer science and additional coursework in other disciplines. The culmination of this experience is a *senior project* course.

The Core

As we considered the current catalog and the current ACM curricular recommendations [2], we decided it was appropriate to move from a 3- to a 4-credit class model with closed labs. Adding the extra hour allowed us to shift course content, which allowed us to cover

a greater breadth and depth in the core of the curriculum. The intent of the core is to prepare students for upper-division work in the discipline. Of particular interest in the model:

- *CS1-CS2* – in the past, our students were asked to learn as many as 4 programming languages in order to get their degree, almost all of which were taught at an "introductory" level. The CS1-CS2 sequence, with lab, will place a focus on students learning to program and learning one language in depth.

- *Computer organization* – the previous curriculum required students to complete both computer organization and a course in assembly language programming. With the addition of the lab, we can introduce assembly language during this course, as well as microprogramming on embedded systems kits and low-level C programming.

- *Systems programming* – the current catalog had required courses in operating systems (CS) and networking (CIS). Content from these courses were merged into a course that will help students learn how to program using systems utilities.

- *Information ethics* – currently, there is not a required course in this area. Given the high-stakes fields in which many of our graduates are employed (esp. the avionics industry), we felt it was critical to include this in the core. The course will be offered during our winter interim term in a seminar format.

The core also includes a traditional course in discrete mathematics, currently taught by math faculty in our department.

Areas of Specialization

One common criticism of computer science on a liberal arts campus is that the field is too technical and/or vocational. While Mount Mercy is currently not a liberal arts college in the traditional sense, it aspires to be in the future. We saw the development of an interdisciplinary curriculum as a way to make our program fit within the liberal arts approach to education. At the same time, we were conscious to not sacrifice breadth for depth.

Working with faculty from departments across the Mount Mercy campus, we have developed five sample areas of specialization:

- *Computational Science* – this is more or less a traditional computer science degree, including a semester of calculus, linear algebra, probability and statistics, programming languages, algorithm analysis, and a topics course in CS.

- *Web Programming* – this program includes courses in art, marketing, and CS courses in web development and databases.

- *Software Development* – this is more or less the analog to the traditional CIS program, with courses from accounting and management, as well as computer science courses in project analysis, project management, and databases.

- *Embedded Systems* – this area was developed with the Rockwell-Collins clientele in mind, and includes courses in embedded systems, robotics, physics, and mathematics.

- *Information Security* – this area is a blend of criminal justice courses, math, and CS courses in cryptography and databases.

We are also proposing the addition of an area in *educational software*, in conjunction with the education department. A Bioinformatics major is also in the works, developed in cooperation with the Biology, Chemistry and Math programs. We believe this extremely rigorous major will provide an outstanding option for our students without a significant staffing issue, since all of the courses in the major are currently offered. Plans for the complete design and implementation of this major have been postponed until the fall of 2009.

A key element in this curriculum is the creation of a structured set of *topics* courses. Our intent is to bring in adjuncts from area firms to teach these courses, thus allowing us offer several high-interest courses that can change over time without putting pressure on our limited resources. We believe that this model will allow us to be responsive to trends in the discipline as well as to the needs of the firms that employ the lion's share of our students.

The interdisciplinary aspect of the model has been well received by our colleagues in business, education, criminal justice, art, biology and mathematics, as evidenced by their willingness to work with us to develop these sample areas of specializations. It has also been well-received by potential students as we talk to them. The fact that students can complete a minor in one of their sub-disciplines by taking one or more additional courses.

We suspect that the majority of students will look at these five to seven sample programs as their *only* options and will choose from them. Our goal, however, is to develop a repository of student-created programs over time. We have developed a reasonably thorough program-approval process, so that students are not able to load up on so-called "easy" classes as they create their program of study.

Portfolios and the Senior Project

Perhaps the best evidence of our program as a *studio* approach is the inclusion of portfolios to be developed by each student. All courses in the curriculum will be project-based, and students will be required to present these projects in increasingly public formats throughout their progression through the curriculum. These projects and the

presentations will be gathered into an electronic portfolio, which will serve two purposes. First and foremost, it will serve as a means of program assessment and improvement for the department. Second, it will provide the student with an artifact that they can use to present evidence of their work and ability to potential employers and graduate schools.

Missing from the current curriculum was a significant capstone experience in the computer science major (most other Mount Mercy programs have a course or series of courses). Our goal with senior project is to give students the opportunity to exercise their skills in their chosen area of specialization. The project will be tailored to that area as much as possible, whether it is the development of an application for an external client, a significant undergraduate research work, or the development of software for some entity within the college. Our goal is to have public presentations of these projects on a regular basis, getting feedback from the community at large (including faculty from other disciplines, area employers, and other students).

Assessment

One of the key pieces of the development of the new curriculum is the development of a comprehensive assessment plan. This is in line with both the overall goals of the college's current administration as well as the Title III Grant. We are in the process of developing course objects that are tied to the ACM's current curricular guidelines [2], the college-wide learning objectives [5], and the objectives that are particular to the college's mission statement (the six *Mercy Concerns* as articulated by the Sisters of Mercy).

As previously mentioned, portfolios of student work are a key component of the assessment of the effectiveness of the new curriculum. We also intend to develop a repository of laboratory activities tied to specific course objectives, for all courses in the core (with the exception of information ethics). Finally, we are in the process of developing an appropriate set of instruments for assessing learning in the core courses, either through a standardized exam (e.g., the Major Field Aptitude Test in Computer Science from Educational Testing Services) or through development of in-house assessments through a vetted process that will include colleagues at other institutions.

Conclusion

The new curriculum was approved by the Mount Mercy College faculty in January 2009, and goes into effect this coming fall. We have already had several students express interest in the program, specifically because of its flexibility. We anticipate developing new and important partnerships with area employers to develop areas of specialization specifically tailored to the needs of their employees.

Obviously, we are not done. As we move forward, we anticipate that there will be numerous learning experiences; these will be cataloged and written up as a follow-up to this work. It is also our intent to gather longitudinal data relative to the effectiveness of this approach to be presented at a future conference. Finally, we look forward to publishing our assessment materials, including objectives, evaluation criteria, and lab exercises in some sort of open format where we can share. We see this as an open process and welcome feedback.

References

[1] Coe College Catalog, 2008-2009.

[2] Engel, G. & Shackelford, R. (Eds.) (2001). Computing curricula 2001: Computer science. New York: ACM Press.

[3] Grand View University Catalog, 2008-2009.

[4] Iowa Wesleyan College Catalog, 2008-2009.

[5] Mount Mercy College Catalog 2008-2009.

[6] Schaefer, J. B. (2005). The Integration of the Software Studio Approach into the Undergraduate Computer Science Curriculum, presented at the Midwest Instruction and Computing Symposium, April 2005.

[7] Simpson College Catalog, 2007, 2009.

[8] Wartburg College Catalog, 2008-2010.

Appendix A – Description of the CS Program and Areas of Specialization

Core Requirements	
MA 150 (Discrete Mathematics)	3 semester hours
CS 105 (Fundamentals of CS)	4 semester hours
CS 106 (Data Structures)	4 semester hours
CS 190 (Computer Organization)	4 semester hours
CS 203 (Information Ethics)	3 semester hours
CS 235 (Systems Programming Concepts)	4 semester hours
CS 435 (Senior Project)	3 semester hours
Total	*25 semester hours*

Area of specialization	
Computer science electives	6-12 semester hours
Specialization courses	11-16 semester hours
Total	*22-23 semester hours*

Grand total	***47 - 48 semester hours***

Areas of Specialization

Computational Science	
CS 302 (programming languages)	4 semester hours
CS 389 (algorithm analysis)	4 semester hours
CS 399 (any special topics course)	3 semester hours
MA 164 (calculus I)	4 semester hours
MA 202 (linear algebra)	4 semester hours
MA 214 (probability and statistics)	3 semester hours
Total	*22 semester hours*

Information security	
CS 315 (web programming)	4 semester hours
CS 399 (special topics in cryptography)	3 semester hours
MA 164 (calculus I)	4 semester hours
MA 165 (calculus II)	4 semester hours
MA 214 (probability and statistics)	3 semester hours
CJ 297 (criminal law)	3 semester hours
CJ 350 (evidence)	3 semester hours
Total	*24 semester hours*

Web Development

CS 315 (web programming)	4 semester hours
CS 388 (database systems)	4 semester hours
BK 208 (principles of marketing)	3 semester hours
BK 300 (marketing communications)	3 semester hours
AR 130 (graphic design I)	3 semester hours
AR 230 (graphic design II)	3 semester hours
AR 285 (microcomputer graphics)	3 semester hours
Total	*23 semester hours*

Software Development

CS 326 (information systems analysis)	3 semester hours
CS 388 (database systems)	4 semester hours
CS 426 (project management)	3 semester hours
BA 250 (tech and comm. in business)	3 semester hours
BC 202 (accounting: info for decisions)	4 semester hours
BK 208 (principles of marketing)	3 semester hours
BN 204 (principles of management)	3 semester hours
Total	*23 semester hours*

Embedded Systems

CS 399 (topics in embedded systems)	3 semester hours
CS 399 (topics in robotics)	3 semester hours
MA 164 (calculus I)	4 semester hours
MA 165 (calculus II)	4 semester hours
PH 151 (principles of physics I)	4.5 semester hours
PH 152 (principles of physics II)	4.5 semester hours
Total	*23 semester hours*

Appendix B – Current Assessment Criteria

This list describes those goals that we will use to evaluate the program as a whole, and defines the means of assessing our performance related to these goals. This is a work in progress. In general, we believe that students graduating from Mount Mercy College with a degree in computer science will be able to:

- Demonstrate an ability to solve problems of varying complexity and content areas.
  - Feedback from employers hiring our graduates as well as those hiring interns
  - Participation in regional programming contests (ACM, MICS)
  - A portfolio of projects gathered from at least 4 courses in the program of study

- Demonstrate an ability to write programs at a professional level.
  - Periodic feedback from employers who hire our graduates
  - Completion of a professional certification exam in a programming language or environment (e.g., Java or MySQL)
  - Evaluation of the senior project

- Demonstrate an ability to work in a team-based environment, assuming multiple roles within a team.
  - Feedback from organizations for whom students work on their senior project
  - Feedback from other team members on the senior project
  - Periodic feedback from employers who hire our graduates
  - Evaluations of team programming projects done in classes at Mount Mercy College

- Demonstrate effective written communication skills in the field of computing.
  - Students taking the information ethics course will present a paper in a public seminar on some topic in information ethics; attendees will be asked to evaluate the presentations using an appropriate form.
  - Students taking senior project will prepare documentation (requirements, specifications, help manuals) for their clients; clients will provide feedback on the quality of the documentation.
  - Students will be encouraged to submit papers to conferences.

- Demonstrate ethical behavior in their personal and professional lives.
  - Feedback from employers hiring our graduates as well as those hiring interns
  - Evaluations of team programming projects done in classes at Mount Mercy College

- Demonstrate a commitment to service to others at work and the community at large.
  - Feedback from employers hiring our graduates as well as those hiring interns
  - Written summaries provided by graduating students of activities done at Mount Mercy College and/or in the Cedar Rapids community during their time here at Mount Mercy College.