MODULAR CURRICULUM DESIGN: A MODEL FOR

ARTICULATED PROGRAMS OF STUDY

Paul M. Leidig
Computer Science & Information Systems
Grand Valley State University
Allendale, MI 49401
(616) 895-2308
leidigp@gvsu.edu

ABSTRACT

As many Computer Science programs struggle with offering a wide array of newer course offerings, concern is raised over the burgeoning list of degrees and courses. This phenomenon is compounded as academic units outside of the discipline attempt to provide students with computing skills necessary for their profession. The concept of offering a standard one-size-fits-all approach in the CS curriculum is rapidly becoming insufficient. In addition, an opportunity exists to tailor a student's course selections to better prepare for a wider variety of careers. Concurrent with an increase in popularity of one of our undergraduate minors, other academic units are requesting courses specifically tailored to their programs. Recognizing a trend of CS programs becoming less autonomous, this paper proposes a method of re-engineering the curriculum to better match a variety of needs. By creating a core module of courses, combined with a superset of courses designed specifically for other academic programs, it becomes easier to provide an articulated path of study that better meets the needs of students in non-computing programs.

INTRODUCTION

Over the past several years we have noticed an increasing demand from other academic units at our university, for clear, specific skills and knowledge sets to be provided via computer science courses. Primary to this trend was an increased demand for microcomputing skills such as spreadsheet analysis, database theory and practice, desktop media, and programming for the Windows environment. This demand became evident with increasing requests (from both students and other academic departments), for more sections of courses in spreadsheets, microcomputer databases, and Visual Basic. Our first response was the creation of a minor in Microcomputer Application, beginning with the Fall 1991 semester. The Microcomputer Applications minor (since renamed Microcomputer Systems) offered core courses in computing concepts, programming, spreadsheets, and databases, along with electives in microcomputer operating systems, communications, expert systems, and desktop media.
The current Microcomputer Systems Minor consists of at least 24 hours, as follows:
All of the following:
CS 150 Introduction to Computing
CS 160 Programming in Visual Basic
CS 231 Spreadsheets
CS 233 Microcomputer Database Management
CS 339 Microcomputer Systems Project
Elective courses (select three of the following courses)
CS 162 Computer Science I (C++)
CS 230 Microcomputer Operating Systems
CS 235 Expert Systems
CS 237 Microcomputer Communications
CS 238 Desktop Media
CS 331 Advanced Spreadsheet Development

This minor has become quite popular with students in several programs, primarily with those majoring in business, public administration, political science, and the health sciences. Sections of these courses quickly fill, with lengthy lists of students requesting closed class permits and additional sections. Enrollment has more than doubled in the past four years, limited only by the lack of resources to staff an adequate number of sections.

As departments evaluated the computing skills of students in their programs, we began receiving requests to tailor courses in the microcomputer systems minor to better meet the needs of specific programs. A potential drawback to modifying courses for differing requests, could be courses that no longer meet the needs of the general population. One alternative would be to offer separate sections of a course specifically designed for different majors. Although this solution appears to answer the needs of multiple departments, it creates a potential scheduling and staffing problem for the computer science faculty. Another potential problem is confusion among students about the difference between sections.

There are several curriculum paradigms that offer suggested patterns as a solution. The IS'97 Model Curriculum [1] specifically address the development of multiple discipline relevant skills. Clear [2] presents several curriculum scenarios that attempt to couple the computing disciplines, in which the Shared Model and Integrated Model provide a mechanism for cross-discipline programs. Lee et al. [3] advocate the need for more discrete programs focused on offering differing graduate profiles aimed at career outcomes instead of learned skills. Integrating programs often take more work then simply modifying an existing program, however in many cases may offer our students and their prospective employers a better education.

This paper proposes a cooperative approach, wherein the Computer Science & Information Systems department and other interested departments jointly offer a minor tailored to specific sets of knowledge areas. With this structure, students would take a common set of core classes to ensure consistency among the various minors, and a superset of classes, some offered by the CS & IS department and others by the cooperating unit. We refer to this approach as an articulated minor. Each articulated
minor would be cross-listed in the computer science program, and also in the associated unit.

A MODEL FOR AN ARTICULATED MINOR

For descriptive purposes of this paper, an articulated minor will be referred to as one in which a student takes a prescribed set of courses in his/her major area of study, as well as a specific set of courses in another field that provides a set of functional skills in addition to the major. By specifying a set of courses in the major area of study, and a complementary set of courses in another area, it is convenient for students in both majors to elect the articulated minor. For example, we offer a minor in Computer Engineering. This minor consists of a core set of courses, plus two separate groups of courses; one group of computer science courses taken by engineering students, and a set of engineering courses taken by computer science majors. Figure 1 illustrates the requirements for a Minor in Computer Engineering. In this example, a student majoring in engineering, must take eight courses; the four courses in the minor core (c) and the four courses in computer science depth courses (a) to receive a minor in Computer Engineering. The sum of the required courses (a, b, and c) therefore make up the minor, although one of the two additional groups of courses (b) would be taken as part of the major.

The Minor in Computer Engineering consists of:

A common core (c)
EGR 226 Intro. to Digital Systems
EGR 326 Adv. Digital Systems
CS 162 Computer Science I
CS 262 Computer Science II

Minor subset (a)
CS 362 Data Structures and Algorithms
CS 451 Computer Architecture
CS 452 Operating Systems
CS 457 Data Communications

Minor subset (b)
EGR 214 Circuit Analysis
EGR 424 Design of Microcontroller Applications

The above minor serves as an excellent example to describe a model to use in the creation of an articulated minor. The application of this model when creating a new
minor will be of assistance in alleviating the problem mentioned in the introduction of this paper, namely the accumulation of a series of disparate and dissimilar minors. This model will be used to redefine the existing Microcomputer Systems minor, and serve as the basis for future similar articulated minors.

**APPLYING THE ARTICULATED MINOR MODEL**

As an example of the application of the proposed model, we will first redefine the Microcomputer Systems minor described in the Introduction. The modified Minor in Microcomputer Systems consists of a rearranged list of courses. The current list of courses has been separated into two groups; those that would constitute a required, or 'core' set of courses, and a set of elective computer science courses. Each minor will be completed with a capstone project course. The revised Minor in Microcomputer Systems consists of:

**Revised Microcomputer Systems Minor**

**Core courses:**
- CS 150 Introduction to Computing
- CS 160 Programming in Visual Basic
- CS 231 Spreadsheets
- CS 233 Microcomputer Database Management
- CS 237 Microcomputer Communications

**Elective courses:** (select three of the following courses)
- CS 162 Computer Science I
- CS 230 Microcomputer Operating Systems
- CS 235 Expert Systems
- CS 238 Desktop Media
- CS 331 Advanced Spreadsheet Development
- CS 339 Microcomputer Systems Project

Drawing from the revised Microcomputer Systems minor, we have now identified a core set of courses that would be required in any similar set of articulated minors. We can define future minors as this core, plus the articulated list unique to that minor. This core group of courses may now be used as the basis for maintaining consistency among any future proposed minors.

Using this model, we can now describe two articulated minors; a Healthcare Information Systems minor offered with the School of Health Professions, and an Accounting Systems minor proposed with the School of Business. Among the various academic units that have requested courses that better meet the needs of their students, these two minors provide good examples of applying this model.

The Articulated Minor in Healthcare Information Systems consists of courses from health science, sociology, computer science, and a capstone project course.

**Articulated Healthcare Information Systems Minor**
- CS 150 Introduction to Computing
- CS 160 Programming in Visual Basic
- CS 231 Spreadsheets
- CS 233 Microcomputer Database Management
- CS 237 Microcomputer Communications
HS 210 Introduction to Health Professions
HS 220 Health Care Delivery
HS 222 Introduction to Public Health
HS 340 Health Care Management
SOC 356 Sociology of Health Care
CS 340 Healthcare Information Systems
CS 493 Healthcare Information Systems Project

The Articulated Minor in Accounting Systems consists of courses from accounting, management, computer science, and a capstone project course. The complete list of courses is shown below.

Articulated Accounting Systems Minor
CS 150 Introduction to Computing
CS 160 Programming in Visual Basic
CS 231 Spreadsheets
CS 233 Microcomputer Database Management
CS 237 Microcomputer Communications
ACC 212 Financial Accounting
ACC 213 Managerial Accounting
ACC 314 Intermediate Accounting
ACC 340 Accounting Systems
ACC 414 Auditing
MGT 368 Management Information Systems
CS 331 Advanced Spreadsheet Development
ACC/CS 495 Accounting Systems project

At first glance, this complete listing of courses could appear quite daunting for a student considering one of these minors. By re-ordering the complete list of courses that make up the minor, into the three sets as described in Figure 1, it becomes easier to visualize and describe the additional requirements for a given major. As an example, a student majoring in Accounting who elects to receive the Accounting Systems minor, could view the program in a modular fashion. Instead of looking at 13 courses as his/her requirements for the minor, there would be three modules in the minor. This model shows three modules: the core with five required courses, six accounting and management courses, and two additional courses tailored for the Accounting Systems Minor.

Modular Articulated Accounting Systems Minor

Core courses in the minor: (c)
CS 150 Introduction to Computing
CS 160 Programming in Visual Basic
CS 231 Spreadsheets
CS 233 Microcomputer Database Management
CS 237 Microcomputer Communications

Courses included in the accounting major. (b)
ACC 212 Financial Accounting
ACC 213 Managerial Accounting
ACC 314 Intermediate Accounting
ACC 340 Accounting Systems
ACC 414 Auditing  
MGT 368 Management Information Systems  
Additional courses from computer science. (a)  
CS 331 Advanced Spreadsheet Development  
ACC/CS 495 Accounting Systems project  
By looking at the modular list of courses, the accounting major will quickly recognize the required courses in the major. Using the model in Figure 1, we will refer to this set of classes as set (b). A review of Figure 1 demonstrates then, the additional requirements of the minor, for the accounting major, becomes a subset of the entire minor requirements. In this example, the additional required courses becomes the combination of sets (a) and (c).  
Articulated Accounting Systems Minor  
Subset for the Accounting Major  
CS 150 Introduction to Computing  
CS 160 Programming in Visual Basic  
CS 231 Spreadsheets  
CS 233 Microcomputer Database Management  
CS 237 Microcomputer Communications  
CS 331 Advanced Spreadsheet Development  
ACC/CS 495 Accounting Systems project  
For the accounting major, the Accounting Systems minor becomes a list of seven additional courses. From an advising standpoint, this becomes an easier minor to explain and track. Perhaps more importantly, this view of the minor is a much more attractive option for the student considering the feasibility of such a program. It should be noted however, that the complete list of courses in the minor will appear in the Requirements for the Minor, and on the students' official transcript.  

IMPLICATIONS AND PRACTICALITY  
Describing and promoting an articulated minor as described in this article offers the same advantages to many academic units. By creating separate minors tailored for specific majors, departments offering traditional information systems or computer science degrees offer attractive alternative educational opportunities. Looking for a solution to requests for more specific knowledge sets in common service courses, we have found not only a way to respond to those requests, but offer a complete minor in many different programs. The two articulated minors mentioned above are only the first in a series of planned minor proposals.  
How practical and attractive is this model? To apply the application of the model to specific majors, we wanted to measure interest in the first such minor being proposed. As part of a proposal for the Minor in Healthcare Information Systems mentioned above, we administered a survey to health science students, the primary target audience for this minor. Responses to the survey indicated that 79.4% of health science students found this an attractive minor. The ability to extend a traditional degree with an articulated minor in a related and complementary field is at least one reason to consider using such a model.
CONCLUSIONS

An opportunity is presented whereby computer science courses serve as the foundation of a model curriculum that caters to the computing technology needs of students in non-computing programs. Offering a standard one-size-fits-all computer science program is often an unattractive option for many students. This paper has proposed a method of re-engineering the curriculum to better match a variety of needs of students outside the discipline. By creating a core module of courses, combined with a superset of courses designed specifically for other academic programs, it becomes easier to provide an articulated path of study that better meets the needs of students. One outcome of this model is a set of additional courses that becomes an attractive enhancement to other non-computer programs.

REFERENCES

