Interdisciplinary Application Tracks in an Undergraduate Computer Science Curriculum

Department Curriculum Committee
Mingrui Zhang, Eugene Lundak, Chi-Cheng Lin, Tim Gegg-Harrison, Joan Francioni
Computer Science Department
Winona State University
Winona, MN 55987
{mzhang, elundak, clin, tgeggharrison, jfrancioni}@winona.edu

ABSTRACT
The Computer Science Department at Winona State University revised its curriculum to include an interdisciplinary approach adapted to the study of computer science. The new curriculum consists of a traditional Computer Science option and an Applied Computer Science option consisting of four separate tracks, namely: bioinformatics, computer information systems, geographic information technology, and human computer interaction. This paper describes the design strategy and implementation plan as well as the content of our multi-track Applied Computer Science curriculum.

Categories and Subject Descriptors
K.3.2 [**Computer and Information Science Education**]: *Computer science education, Curriculum, Information Systems education*

General Terms
Design, Experimentation

Keywords
Applied Computer Science, Interdisciplinary, Multi-Track Curriculum, Bioinformatics, Computer Information Systems, Human Computer Interaction, Geographic Information Systems, Geographic Information Technology

1. INTRODUCTION
Computer Science (CS) is not only an area of study in its own right but it is also an important supporting area for many other disciplines. As such, the definition of computer science is broadening to include areas of emphases directly related to other disciplines. Not only is this beneficial for the computer science discipline [5], it is also beneficial for students to have the opportunity to study another discipline in the context of computer science.

SIGCSE'07, March 7–10, 2007, Covington, Kentucky, USA.

Despite the growth in the application of computers, at a national level, computer science has seen a decline in undergraduate enrollments and an increased migration out of the major since 2000 [4, 11]. One factor in this is the increase in non-CS programs that integrate computer information technology into their curriculum. As such, students are able to learn to use computer technology in the context of other disciplines in which they may have more interest. In these programs, however, students do not typically learn enough about the discipline of computer science to either work in or contribute to the field as "computer scientists."

Is it possible, and is it our responsibility as Computer Science educators, to design curricula where CS students directly learn how to apply their computer science content knowledge to an application area? Morris and Lee of Carnegie Mellon University argue that it is critical for us to "broaden computer science curriculums" and spend more effort teaching students how computing fits into the world than we have been doing [8]. In doing so, they claim we will be able to "prepare our students for more wide-ranging careers." As early as 1993, studies have shown that failure to motivate interest in science by establishing its relevance to the students' lives and personal interests was a significant factor for students who had an initial intention and the ability to major in a science field but instead switched to nonscientific fields [10]. In addition, there is strong evidence that making explicit connections for students between computer science content knowledge and real world examples can increase retention of female students in computer science [7].

These factors, along with our university's interest in developing more interdisciplinary programs across campus, led our department to first consider developing interdisciplinary options for our majors in 2003. In the Spring of 2005, a new curriculum featuring an *Applied Computer Science* option within Computer Science was formally approved and put into place in the Fall 2005 semester. How we moved from initial discussions to a final curriculum, and how the program is doing one year later, are the topics of this paper.

As we started our discussions, many questions naturally arose. What goals do we want to accomplish? What application fields should be included in the new curriculum? Does the department have the expertise needed to address the applied areas? How do the changes impact other academic programs in our university? And, significantly, how will the new curriculum be assessed? In this paper, we discuss the process that eventually worked for us to answer these questions and the resulting applied tracks that were developed and integrated into our traditional computer science curriculum. In Section 2, we present the design of the new curriculum. Section 3 outlines the details of each individual track, while Section 4 addresses four specific implementation issues. We summarize and report on the "state-of-the-program" after one full year in Section 5.

2. DESIGN OF NEW CS CURRICULUM

Our goal for the new curriculum was to provide options within our computer science program where students could choose to study and apply computer science concepts in the context of other disciplines. At the same time, we wanted to maintain the integrity of the two traditional options in our existing curriculum – Computer Science and Computer Information Systems.

To begin with, we did background research into mature and successful application areas of computer technologies, CS curricula at other undergraduate institutes around the nation, and both the national and regional needs of information technology professionals. A number of emerging and exciting candidate application fields were identified, including bioinformatics, computer security, information assurance, scientific computation, computer gaming, human computer interaction, and geographic information systems. Based on our current CS curriculum structure, our faculty expertise, and the support and interests from other academic programs and local businesses in the region, we eventually chose to keep two separate options in the program and name them "Computer Science" and "Applied Computer Science." Within the Applied Computer Science (ACS) option, we then designed four applied tracks: Bioinformatics, Computer Information Systems (CIS), Human Computer Interaction (HCI), and Geographic Information Technology (GIT). One common factor determined for all the tracks/options was that they would offer full-fledged Computer Science degrees and they would share a common core of courses. Figure 1 shows a diagram of our structure.

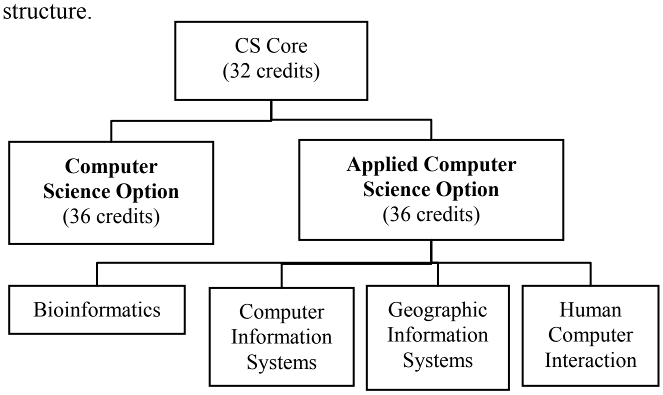

Figure 1. Multi-track Computer Science Curriculum at Winona State University

Our approach to designing the new curriculum from that point on was both bottom-up and top-down: the design of each track was bottom-up; the structural realignment of the tracks within the curriculum was top-down. After the tracks were determined, four track specific task forces and one core task force were formed to look into the plans for adapting them. Each track task force was charged with interacting with the impacted academic programs in the university, determining an appropriate set of track specific curriculum components, and then to making recommendations to the core task force. The objective was that the courses for each track would form an integrated aggregate instead of simply "a bag of courses."

2.1 Computer Science Core

The core task force coordinated the integration of the tracks into the curriculum. It also realigned the curriculum components of the proposed applied tracks and existing courses in such a way that the core set of courses in the new curriculum would comply with the recommendations of Computing Curricula 2001 [1, 2]. The resulting CS Core requirement, listed in Figure 2, totals 9 courses of 32 semester credits and covers the core computer science knowledge units recommended in CC2001.

Algorithms and Problem Solving I (4 credits)
Algorithms and Problem Solving II (4 credits)
Discrete Mathematics for Computer Science (4 credits)
Social Implications of Computing (3 credits)
Data Structures (4 credits)
Computer Systems (4 credits)
Applied Database Management Systems (3 credits)
Software Engineering (3 credits)
Object-Oriented Design and Development (3 credits)

Figure 2. Core Requirement of Computer Science Curriculum

2.2 The Applied Computer Science Option

Given the number of credits allotted to the CS Core, 36 credit hours were available for each track. In addition, we made the decision to require certain General Education courses in each track. This was a bit of a gamble, as the university-level curriculum committee needed to agree with this. But, given discussions with members of that committee and the university's interest in developing new interdisciplinary programs, we felt this would work. In doing so, an added benefit to our tracks is that students can easily complete a minor in the applied area without taking more than the required 128 credits to graduate while still meeting all of their general education requirements.

2.2.1 Common Framework for Applied Tracks

After several iterations of proposed tracks from the individual tasks forces, members of the entire curriculum committee eventually decided on a common framework for the applied tracks. This framework is outlined in Figure 3.

General Education	Track relevant courses.
Mathematics & Statistics	Track appropriate Calculus; Track appropriate course.
Computer Science	At least 5 courses; at least 3 at the senior level.
Applied Area	Coherent set of non-CS courses
Electives	At least one elective in either CS or the applied area.
Bridge Courses	At least two courses that make explicit connections between computer science and the applied area.

Figure 3. Applied Track Framework

Working within this framework, each track task force adapted track specific components to assure the cohesion of the whole program. Specifically, at least two *bridge courses* were defined for each track that would make explicit connections between computer science and the applied area. As such, these courses were expected to have pre-requisites in both areas.

When designing a track, it was important to realize that students graduating from our program may choose to follow a computer science career or go on to graduate school in computer science or a related field. Thus, each track needed to offer students a chance to study particular topics in more depth, thereby increasing not only their knowledge, but also their problem-solving skills as they worked on harder problems. We expect our students to be competitive both for general IT jobs and also jobs specifically relevant to the track. These were difficult tasks to achieve without increasing the total credit hours necessary for a student to graduate from the track. This was particularly difficult for the Bioinformatics track, as a deep understanding of bioinformatics requires a great deal of bio-molecular background.

2.2.2 Inclusion of General Education Requirements

A challenge for designing each track was to pick courses that could help students understand sufficient background about the applied field without significantly increasing their total credit hours. At WSU, students need to complete 128 credit hours before they graduate, where 46 of them come from the general education requirements other than CS. Thus, for each track, a set of general education courses covering the necessary knowledge units for the track has been identified. For example, we require students in the Bioinformatics track to complete a total of 21 credit hours of Biology and Chemistry courses: 14 of them are counted toward their track credits and the other 7 credits are used to satisfy their general education requirements.

An added benefit to this approach is that incoming students are able to "try out" a track by taking relevant track general education courses concurrent with their introductory computer science courses. If they realize right away that they do not like an applied area, they can either switch to another track or to the computer science option without any delay in graduation. The "tried" course, however, still counts towards their general education requirements.

2.2.3 Computer Science Integrity

From the beginning of this process, as a department we were committed to ensuring that the new curriculum be, first and foremost, a "computer science" curriculum. Therefore, we included several levels of quality control into the ongoing process of our curriculum re-design. Our core task force closely examined the Computing Curricula 2001 report to identify all necessary CS knowledge units to include in the new curriculum. Each track task force interviewed impacted departments within the university to identify knowledge units necessary to each track. Information was constantly exchanged between the department faculty, the track task forces, and the core task force. Taking a combined top-down and bottom-up approach, after many iterations we were able to present a final curriculum to the department that included multiple application tracks and still upheld our computer science standard.

3. SPECIFIC TRACK REQUIREMENTS

The course requirements of the individual ACS tracks beyond the core courses defined in Section 2.1 are described in the following subsections.

3.1 Bioinformatics

Bioinformatics is the study of using computational tools and computer technologies to model, analyze, store, retrieve, manage, present, and visualize biological data. Primarily, the data to be processed are huge amounts of molecular biology data, such as DNA sequences and proteins. Our curriculum for this track is modeled off of the Wright State program [6]. Two new courses, *Introduction to Bioinformatics* and *Algorithms in Bioinformatics*, serve as the bridge courses for the Bioinformatics track.

Table 1. Bioinformatics Track Course Requirements

Mathematics & Statistics	▪ Calculus I ▪ Statistics
Computer Science	▪ *Introduction to Bioinformatics* ▪ *Algorithms in Bioinformatics* ▪ Advanced Database Systems ▪ Two CS elective courses
Applied Area	▪ Basics of Life ▪ Organismal Diversity ▪ Genetics ▪ Principles of Chemistry I & II
Track Elective	▪ One of the following courses: Organic Chemistry An upper-level CS elective course

3.2 Computer Information Systems

Computer Information Systems (CIS) involves the study of business-related processes and software. It evolved from the previous CIS option by adding a required component consisting of web programming – both client-side and server-side technologies, and also component-based reusable software architectures. These topics are important in the development of software to support E-business applications. The two courses, *Internet/Web Development* and *Component-based Software Architectures*, serve as the bridge courses for the CIS track.

Table 2. CIS Track Course Requirements

Mathematics & Statistics	▪ Applied Calculus ▪ Statistics
Computer Science	▪ *Internet/Web Development* ▪ *Component-based Software Architectures* ▪ Advanced Database Systems ▪ Two CS elective courses
Applied Area	▪ Financial Accounting Principles ▪ Managerial Accounting Principles ▪ Management Information Systems ▪ Principles of Microeconomics
Track Elective	▪ Three of the following courses: Legal Environment of Business Principles of Macroeconomics Corporate Finance Principles of Marketing Operations Management Organizational Dynamics

3.3 Geographic Information Technology

Geographic Information Technology (GIT) involves the development of software to support the rapidly growing field of Geographic Information Systems (GIS). GIS uses digital technology to combine maps with computer databases. As such, GIS has a broad range of applications that allow businesses, engineers, and governments to make timely and informed decisions. Our GIT track focuses on the spatial information processing and database management aspects of GIS and is based on recommendations from the GIS&T report [9]. The two bridge courses for the GIT track are *Spatial Information Processing* and *Algorithms in Geographic Information Technology*.

Table 3. GIT Track Course Requirements

Mathematics & Statistics	▪ Calculus I ▪ Statistics
Computer Science	▪ *Spatial Information Processing* ▪ *Algorithms in GIT* ▪ Digital Image Processing ▪ Two CS elective courses
Applied Area	▪ Dynamic Earth ▪ Earth & Life through Time ▪ Watershed Science ▪ Field & Analytical Methods I
Track Elective	▪ Three of the following courses: Surficial Processes & Soils Environmental Science GIS Advanced Geomorphology Applied Hydrogeology Global Climate Change Cartography

3.4 Human Computer Interaction

Human Computer Interaction (HCI) is an interdisciplinary field that attempts to understand the tendencies and limitations of humans in order to design and develop effective software that is user friendly [3]. As such, a successful HCI computer scientist must be well-versed in both computer science and psychology. The two courses, *Introduction to Web Programming* and *Human-Computer Interaction*, are the bridge courses for the HCI track.

Table 4. HCI Track Course Requirements

Mathematics & Statistics	▪ Calculus I ▪ Statistics
Computer Science	▪ *Introduction to Web Programming* ▪ *Human-Computer Interaction* ▪ Three courses of CS Electives
Applied Area	▪ General Psychology ▪ Human Factors Psychology ▪ Cognitive Psychology ▪ Sensation and Perception ▪ Introduction to Sociology ▪ Design of Samples and Surveys
Track Elective	▪ Two courses from Graphic Design, upper-level CS, and/or upper-level Psychology courses

4. IMPLEMENTATION ISSUES

In order to make the transition to the new curriculum as smooth as possible, we attempted to identify and address potential roadblocks up front. Our four main concerns included: 1) the integrity of the program, 2) effective use of limited resources, 3) expertise of the faculty, and 4) the impact of the new curriculum on 2+2 and transfer students.

4.1 Program Integrity

The primary concern that departments address every time they modify their curriculum is that the new curriculum maintains the integrity of the program. That concern was especially prevalent for our latest change, as we did not want the introduction of *applied* computer science tracks to erode the *computer science* aspect of the program. In order to address this issue, we expanded the core courses that are taken by CS and ACS students alike to cover all of the required CC2001 topics. In doing so, we not only ensured the integrity of the ACS major, but we also strengthened the CS major.

4.2 Limited Resources

Resource limitations are always a concern, particularly at public higher education institutions that have been experiencing sharp declines in funding. Given that additional funding was not an option, we needed to find ways to leverage the limited resources within the department so as to not interfere with the quality computer science education that we offer. We employed two strategies to address this issue: 1) jump start the ACS curriculum by deploying the new tracks selectively; and 2) design each track's CS bridge courses in such a way that they can also be taken by students outside of the given track.

Each track has two bridge courses. The lower level bridge courses were designed to be generally applicable to both other CS/ACS majors and also CS minors who wish to relate the material to their fields of study. Because the upper-level bridge courses are less applicable to students outside the track, we have attempted to merge them with related courses until the given track grows to be self-sufficient. As an example, during the Fall semester of 2006, we offered the *Algorithms in Bioinformatics* course concurrently with the standard *Design and Analysis of Algorithms* course by scheduling it with a common two-hour lecture and then separate, domain-specific, one-hour lectures.

We also decided to start with three tracks and then phase in the fourth track at some point in the future. Historically, CIS has been a popular option, so we wanted to continue to offer it, albeit as a track within the ACS option now. With our faculty expertise in AI, user interfaces, computer graphics/gaming, and computer accessibility, HCI was also a natural choice for the initial phase. Due to close proximity and faculty connections to the Mayo Clinic in Rochester, MN, a world-renowned biomedical research center, we decided to implement the Bioinformatics track as well.

4.3 Faculty Expertise

There are four primary sources for expertise from which to pull to support applied tracks such as ours: 1) existing faculty who already have the expertise within the department, 2) individuals outside the department who can provide the expertise, 3) existing faculty who gain the expertise by profession development such as taking classes and attending workshops, and 4) new faculty who could fill certain voids in expertise. We have exploited each of these avenues. For our HCI and GIT tracks, we already had faculty with the relevant expertise. For Bioinformatics, on the other hand, we had to acquire that expertise. The faculty members in the Bioinformatics track task force have been committed to studying molecular biology, genetics, and genomics, and to developing their expertise in bioinformatics. They have also fostered and maintained an ongoing relationship with researchers and bioinformaticians at the Mayo Clinic for the sustainability of the track. Finally, the department is currently in the process of hiring a faculty member with expertise in bioinformatics.

4.4 2+2 and Transfer Students

Winona State University maintains a strong partnership with a regional community college. The two institutions have set up several 2+2 articulation agreements in various academic programs, including both CS and CIS, with the goal to facilitate the transfer process between them. Typically, students in the 2+2 program finish the first two years' study at the community college and then transfer to WSU to complete their study for the B.S. degree. Under the 2+2 agreements, the undergraduate general education requirement is fulfilled once the student receives the A.S. or A.A. degree from the community college. Thus, the 2+2 students can focus their study on courses in their majors in the final two years. The situation is very similar for many transfer students who arrive on campus with courses taken at other institutions. Transferring is never seamless, but we believe that the transition to our program should be reasonably smooth given our expanded set of core courses that now cover all of the required components of CC2001.

In order to maintain our partnership, we met with the administrators and faculty at the community college to revise and extend the existing 2+2 CS and CIS agreements. In addition to revising the CS and CIS agreements, we decided to create just one new collaboration in Bioinformatics, leaving HCI and GIT for possible future expansion. After a series of inter-institutional meetings, including curriculum coordination and the validation of the state's education policy compliance, a new 2+2 articulation agreement in Bioinformatics between the two institutes was finalized in summer 2006. The community college now has a new A.S. degree in pre-Bioinformatics approved by the state's higher education curriculum board.

5. CONCLUSION AND CURRENT STATUS

Although at times it was a difficult process, our department was able to successfully develop a new curriculum that offers students a number of computer science programs with an interdisciplinary focus. Our final design includes a well-defined set of required CS courses to ensure an appropriate coverage of core computer science concepts, along with options for students to take either a traditional set of computer science courses or a cohesive set of computer science and applied discipline-specific courses. Full details of this curriculum are available on the department website: http://cs.winona.edu.

Our new curriculum went into effect in the Fall of 2005. As of Fall 2006, the breakdown of the number of students currently declared as a Computer Science major is shown in Table 5. These enrollment numbers include the students already majoring in either the CS or CIS option in the old curriculum. Although very preliminary, we are encouraged by these numbers. From 2005 to 2006, there is a relatively stable enrollment in the CS Option. However, there is evidence that students who might have traditionally chosen CIS over CS are now experimenting with the other applied areas. Based on discussions with our incoming students, it appears that these tracks are more interesting to them than CIS.

Table 5. Enrollments by Tracks

	Fall 2005	Fall 2006
CS Option	102	105
Bioinformatics	6	10
CIS	64	52
GIT*	0	0
HCI	10	20
Total	182	187

*Since we have not yet started the GIT track, that enrollment is still at 0

Assessment of our program, in the short term, will focus on our retention rates in the applied tracks. We are also tracking students' choice of major to find out if they change to a different track, if they change to a different major altogether, and when they make this decision. In the long term, we will be able to assess the kinds of jobs our graduates take and their preparedness as well as their interest for the job.

6. REFERENCES

[1] ACM/IEEE Joint Task Force on Computing Curricula (2001). Computing Curricula 2001: Computer Science. http://www.sigcse.org/cc2001/.

[2] ACM/IEEE Joint Task Force for Computing Curricula (2005). Computing Curricula 2005: The Overview Report. http://www.acm.org/education/curricula.html#CC2005.

[3] ACM SIGCHI Curriculum Development Group (1996). *Curricula for Human-Computer Interaction*. http://www.sigchi.org/cdg/.

[4] Cohoon, J. M. and L. Chen, "Expending the Pipeline: Migrating Out of Computer Science," Vol. 15, No. 2, pp. 2-3, Computing Research News, March 2003.

[5] Denning, P., "Crossing the Chasm: The Keys to a Coherent Profession are Bridges between Computing Technologists and the Multitude," *Communication of the ACM,* Vol 44, No. 4, 21-25, April, 2001.

[6] Doom, T., M. Raymer, D. Krane and O. Garcia, "Crossing the Interdisciplinary Barrier: A Baccalaureate Computer Science Option in Bioinformatics," *IEEE Transactions on Education*, 46(3): 387-393, August 2003.

[7] Fisher, Allan and Margolis, Jane, "Unlocking the Clubhouse: The Carnegie Mellon Experience," Inroads - SIGCSE Bulletin, Special Issue: Women and Computing, Vol 34, No. 2, June 2002, pp. 79-83.

[8] Morris, James, H. and Lee, Peter, "The Incredibly Shrinking Pipeline Is Not Just for Women Anymore," in Computing Research News, May 2004, Vol. 16, No. 3.

[9] Task Force on the Development of Model Undergraduate Curricula, "Development of Model Undergraduate Curricula for Geographic Information Science & Technology – The Strawman Report," *University Consortium for Geographic Information Science*, June 2003.

[10] Tobias, Sheila. *They're Not Dumb, They're Different: Stalking the Second Tier.* Research Corporation, Tucson, 1990.

[11] Vegso, Jay, "Drop in CS Bachelor's Degree Production," in Computing Research News, March 2006, Vol. 18, No. 2.