Computer Science Curriculum Revision

Background

Undergraduate programs in computing-related disciplines began to emerge in the 1960s. At that time there were only three kinds of computing-related programs: computer science, electrical engineering, and information systems. Each program covered a well-defined domain area within computing. For students interested in developing software or in the theoretical aspects of computing, computer science was the obvious and only choice. As the computing field matured and grew, new computing related disciplines emerged. Computer engineering started to emerge from electrical engineering in the 1980s. Around the same time, software engineering and information technology emerged from computer science as domain areas within the computing field and began to develop into disciplines. (The Joint Task Force for Computing Curricula, 2005)

The development of the various computing programs at RIT followed a similar pattern. A proposal for a BS degree in Computer Technology in 1967 represented RIT's entry into the computing field. The degree program described in the proposal consisted of courses from the first two years of the engineering program with additional courses in theory, algorithms, numerical analysis, system programming, and formal languages. In the late 1960s and early 1970s the academic computing program and RIT's computing services were administered together. In October 1971, a new academic unit called the School of Computer Science and Technology (CS&T) was formally proposed. It consisted of two departments: Computer Engineering and Computer Systems/Programming. In the fall of 1973 CS&T was administratively moved into a newly formed RIT college, the Institute College. By the end of the 1974-1975 academic year recognizable Computer Science and Computer Engineering programs were established at RIT.

As the computing landscape changed, RIT changed as well. In 1989 the Department of Information Technology was created and in 2001 the Department of Software Engineering was created. As RIT's computing programs grew it became apparent that the computing programs on campus lacked a clear identity. A proposal for the formation of a new college of computing was developed, and on July 1, 2001 the B. Thomas Golisano College of Computing and Information Sciences (GCCIS) was formed. Today GCCIS is one of the largest comprehensive computing colleges in the country and offers a rich and diverse set of computing-related degree programs that meet the needs of virtually every aspect of the computing field. New departments and programs continue to emerge.

The current computer science undergraduate curricular structure (as reflected in the current BS degree requirements) has been used with minor modifications for the better part of a quarter of a century. During that timeframe, course content has changed, and courses have been added (or deleted) to reflect the expertise of the faculty and the expected interest on the part of students. However, the basic philosophy of the curriculum has remained the same even though the discipline and, more importantly the structure of the college, changed dramatically. The growth and diversification of the programs within GCCIS has created an opportunity for the Computer Science Department to reassess its role in the College. As the departments in the college offer new degree programs based on topics traditionally taught in computer science, the department can redefine and sharpen its focus to ensure that it continues to provide its students with the best possible education in computer science.

Two years ago the department initiated a formal curricular review process that included a review of assessment data collected over the past few years and a study of curriculum ideas developed

at other schools and by professional computing organizations. The goal of this effort was to modernize our BS curriculum, set our program apart from other Computer Science programs in the country, and from other computing programs offered at RIT. The result of this study is a proposal for a new curriculum for the undergraduate BS degree in Computer Science (Ad Hoc Curriculum Committee, 2008). The proposal was approved by the faculty at its May 2008 faculty meeting and is described in this document.

Overview of Changes

Structurally, we propose modest, yet important changes to the curriculum. There are changes in Mathematics and Liberal Arts/General Education designed to better serve Computer Science majors in the future. In order to make more room for computer science courses, we have moved coverage of technical communication and ethics to the general education category. A second software engineering course has been added to the core to provide students with a more robust background in modern software engineering practices. In order to ensure that our students have both breadth and depth in advanced computing topics, we require students to take only computer science courses to satisfy computer science electives and insist that students choose a minimum set of courses from one topic area. Finally, we allow students greater flexibility of choice with respect to Free Electives. Students can use their free electives to take virtually any course offered on campus giving them adequate room in their schedule to pursue minors. A comparison of the old and new curriculum is given in Table 1.

The most significant changes occur in what is termed the core or required computer science courses. The goal of the redesigned core is to provide an early and thorough introduction to problem solving before moving on to study practical applications of problem solving using several different programming languages. In addition to changes in content, a realignment of topics provides students with a better view of modern computing systems, paradigms, and issues. The core can be divided into two parts: foundation and breadth. The foundation courses are designed to give students an overview of the entire field of computer science coupled with proficiency in the basic skills of computer science, namely problem solving and algorithm development. The foundation courses consist of: “Invitation to Computer Science,” “Problem-Based Introduction to Computer Science,” “Data Structures for Problem Solving,” “Object-Oriented Programming,” and “Imperative Programming.”

The course titled “Invitation to Computer Science” has the primary goal of providing an early introduction to the entire field of computer science so that students do not get the mistaken idea that computer science is only about programming. The foundation courses, “Problem-Based Introduction to Computer Science” and “Data Structures for Problem Solving,” are designed to provide an introduction to basic computer science concepts through the study of computational problems and their solutions, including the use of a programming language to develop concrete solutions. These courses introduce both the imperative and functional programming paradigms. Functional programming is increasingly receiving attention as a way to program modern multi-core computers. Finally, these courses introduce program testing and correctness early in the curriculum as central rather than peripheral aspects of implementation.

The breadth courses are designed to provide students with a deeper exposure to fundamental topics in computer science. The breadth courses consist of: “Concepts of Computer Systems,” “Concepts of Parallel and Distributed Systems,” “Introduction to CS Theory,” “Introduction to Intelligent Systems,” “Concepts of Data Management,” and “Analysis of Algorithms.” Courses in computer systems (computer organization and architecture) and computer science theory are part of the current core of the BS program. The addition of an algorithms course to the core reflects the desire of the department to create a program that embraces the fundamental science of computing.

Category	Current	Proposed
CS Required	40 QH	42 QH
CS Electives	24 QH	24 QH
Software Engineering	4 QH	8 QH
Mathematics	24 QH	28 QH
Science	20 QH	20 QH
Liberal Arts	52 QH	36 QH
Related Electives	12 QH	--
Free Electives	12 QH	20 QH
Additional Liberal Arts Courses	--	8 QH
FYE/Wellness	2 QH + 2 Activities	2 QH + 2 Activities
Co-op	4 blocks	4 blocks
Totals	**190 QH + Wellness/Co-op**	**188 QH + Wellness/Co-op**

Table 1 - Structural Comparisons

Modern computing hardware consists of multi-core CPUs, or clusters of machines connected by a local network, or distributed systems connected over the Internet. All computer science students must understand how these parallel and distributed systems work and must be able to work with them. The course, “Concepts of Parallel and Distributed Systems” has been added to the core to ensure that students have the basic knowledge required to effectively work with this equipment. In a similar light, virtually every modern software system makes use of a database to organize and store data. In the modern data driven world, computer scientists must understand the basic concepts of database systems. Effective data management is a vital tool in the toolbox of any modern problem solver. To ensure that our students are well versed in the basic application and theory of database systems, the course “Concepts of Data Management” has been added to the core.

Proposed Timeline and Implementation Issues

It is our intent to have the new curriculum start in the fall quarter (20101). Our goal is to deliver the curriculum proposal to the GCCIS curriculum committee no later than the start of the winter quarter (20092). We assume that the changes being made to the curriculum are significant enough to mean that the proposal that will need to go through the relevant curriculum bodies at RIT, and then on to the state of New York for final approval. Our undergraduate program coordinator will serve as the coordinator for this project.

Clearly a change of this magnitude cannot be made without consulting other departments in the Institute. We have had preliminary discussions with representatives from the Mathematics, the Philosophy, and the Software Engineering departments. We will follow up with them and request letters of support for this proposal. The proposal has also been strongly endorsed by the Bioinformatics department.

Once the proposal has been completed and on its way through the curricular approval process, we will develop a transition plan for students currently enrolled and make recommendations for how to map courses already taken to the new program. We will also develop recommendations for when to phase out existing courses that would not be a part of the new program. We believe that the foundation courses will satisfy the need of those departments whose students are required to take the current CS introductory sequence. If this is not the case the department is prepared to provide service courses to serve the needs of these departments.

Bibliography

Ad Hoc Curriculum Committee. (2008). *Report from the Ad Hoc Curriculum Committee NG to the CS Faculty.*

Brooks, F. (1996). The Computer Scientist as a Toolsmith II. *Communications of the ACM , 39* (3), 61-68.

Denning, P. J. (2008). *The Computing Field: Structure.* Naval Postgraduate School.

Rosenbloom, P. S. (2004, November). A New Framework for Computer Science and Engineering. *Computer* , 21-36.

Stratton, W. J. (2003). *A Brief History of Computing Programs offered at the Rochester Institute of Technology.* Laboratory for Applied Computing.

The Joint Task Force for Computing Curricula. (2005). *Computing Curricula 2005: The Overview Report.*