Toward a Threads-Based Computing Curriculum

Ju An Wang and Kai Qian
Southern Polytechnic State University
1100 South Marietta Parkway
Marietta, GA 30060-2896, USA
01-678-915-3718
{jiang, kqian}@spsu.edu

ABSTRACT
The threads-based computing curriculum \([1, 2, 3]\) is a compelling education model for undergraduate computing degree programs. The primary goal of a threads-based curriculum is to create a contextualized computing education in which multiple cores, rather than one single core or "one size fits all" curriculum, expand the skill sets of students and motivate students by selecting their career trajectories towards a professional career. This paper discusses some of the characteristics and benefits of a threads-based computing curriculum, as well as practical considerations involved in the introduction and implementation of such a curriculum model in computing schools or departments.

Computing has become an enabling technology for other sciences and technologies. Advances in computing have been the principal drivers of significant economic growth worldwide. However, as NSF \([4]\) pointed out, despite the deep and pervasive impact of computing and the creative efforts of individuals in a small number of institutions, undergraduate computing education today often looks much as it did several decades ago. We have to revitalize our undergraduate computing education to generate a U.S. workforce with computing competencies and skills imperative to the Nation's health, security, and prosperity in the twenty-first century. It has become increasingly clear that old computing curricula were not effective in generating graduates who remain competitive in an increasingly global, interconnected economy. We must revise and re-design our computing curricula to address the common challenges such as fluctuating enrollments in traditional computer science programs, changes and trends in workforce demographics, the imperative to integrate fast-paced computing innovations into curriculum, and the need of a diverse, agile workforce with the computing knowledge essential to U.S. leadership in the information warfare and global innovation enterprise.

At the same time, there have been several studies exploring how computer science is perceived by students. For instance, Microsoft [5] found out that in some schools, students complained about spending so much time learning about the principles of programming before actually applying them. The integration of principles to solve more tangible problems was often deferred until later, delaying the illustration of relevance. On the other hand, expertise in computer programming is clearly no longer enough to be competitive. What we need is a curriculum model suitable for the flattening world \([8]\). Our computing graduates must be extremely adaptable and become harder to be outsourced.

There are basically two career choices for our students: industry or academia. For career-oriented students, they want to spend more time on learning new skills and new technologies. For students who want to go further study in graduate schools, they need prepare themselves a balanced knowledge set on theory and practical skills. As computing is integrated into many other disciplines and fields, we have to make our students understand the importance of computing in the context of other fields like health sciences, bioinformatics, computational linguistics, astronomy, digital forensics, and a number of other disciplines. More importantly, we could design our computing curriculum in such a way to reflect and implement the immersive and ubiquitous feature of computing.

Threads-based computing curricula recognize the relevance of computing to other disciplines and fields. In a pioneering effort, The College of Computing at Georgia Tech has been working over the past several years to develop threads-based undergraduate computing curriculum, which promises to empower students to continuously add value throughout their careers, be adaptive to changing information technology value propositions, avoid being outsourced, and possess a global mindset \([1, 2, 3]\). The School of Computing and Software Engineering at Southern Polytechnic State University has started \([6, 7, 8]\) to evaluate and adapt the Threads model in our computing curriculum. The major goals of this project include: (1) to assess the application of Threads and its supporting programs developed by Georgia Tech; (2) to study the application of the Threads model to diverse computing programs including Computer Science, Software Engineering, and Information Technology; and (3) to disseminate the results, experiences assessment tools, software support infrastructure, and development process to the larger computing community. With our initial study, we found the Threads model is a great tool to revitalize computing curriculum. We are introducing a security thread in our computing curriculum along with a couple of innovative threads eDevice and Intelligent Computing. We believe that the threads-based curriculum will increase the value of our undergraduate computing degree programs, producing graduates who will be in high demand and who will continuously contribute value throughout successful careers.

The original threads model has two components: (1) students’ computing identities, which are defined by two intertwined pathways through the program, and (2) students’ computing trajectories, which represent what they want to become or how they want to apply their knowledge in the real world. Any two threads can be intertwined, leading to a degree in computing, as presented in the following formula:

\[ \text{Thread}_1 \times \text{Thread}_2 \rightarrow \text{B.S. Degree in Computing} \]

Threads combine regular computer science instruction with additional classes related to a particular area of application, targeting a real-world computing opportunities for a computing graduate. Threads offers students the opportunity to pursue and prepare themselves for a great variety of computing areas, such as
gaming and entertainment computing, intelligent robotics, multimedia and animated movies, mobile computing, bioinformatics, and so on. In the original threads model description [2], four Roles were defined: Master Practitioner, Entrepreneur, Innovator, and Communicator. Threads embody a set of broad, horizontal skills that live within and outside of computing. So graduates from threads-based curriculum have a broad set of skills and are not easily outsourcable.

A thread serves as a context for interpreting the courses in a curriculum for both students and faculty. A thread makes its set of courses cohesive and suggests a coordinated path through its courses so that the end result is expertise in the area of the thread. Each thread is about two-thirds of a degree, but any pair of threads yields a complete bachelor of science degree.

To successfully introduce a threads-based curriculum, we identified four phases in a process of designing and implementing a threads-based curriculum. (1) Design phase: This phase produces a number of “threads” appropriate for the program, supporting the educational objectives and the program outcomes. The design phase includes discussions with students, faculty, employers, graduate schools, target industry and other program constituents about the Threads model and briefing administrators about the threads model to seek their support. Each of our computing programs has an industrial advisory board, and they provide valuable input to our curriculum revise and improvement. Our alumni board has been contributing a great deal to our curriculum design and enhancement as well. After the conceptual work of possible threads is completed, the necessary documentation will be filed for the curriculum committee to approve. (2) Implementation phase: This phase involves formal course and curriculum development. The output of this phase is a list of developed courses for each thread in the threads model. A process and procedure have to be in place for updating and modifying threads as needed. The threads curriculum has to merge with the academic planning process or strategic planning process. It must be consistent with the long-term schedule and required resources must be obtained to implement it. (3) Advising phase: Advising is one of the most important services faculty can provide to students to make smooth transition from traditional curriculum to a Threads-based curriculum. Our project aims at a systematic way to advise our students in the development of meaningful educational plans and achievement of personal and educational goals. We will use software tools, e.g., Threadspace, and develop necessary advising infrastructure to help students benefit from the new threads curriculum model. (4) Assessment phase: This phase includes the development of appropriate evaluation instruments and a process for carrying out such evaluation and assessment and feedback mechanisms to improve the Threads model continuously. We will conduct formal evaluation of our project in a number of areas including academic performance, intellectual growth, student engagement, enrollment, retention, and whether Threads better prepare students for their future careers.

As discussed in [6, 7], the threads curriculum model represents a tremendous departure from how people think about curricula in computing, a departure from a vertically-oriented curriculum whose goal is the creation of students with a fixed set of skills and knowledge. The threads model does not following the old way of monolithic core curriculum plus a number of elective courses for students to choose. This significant change brings in many academic as well as administrative challenges and consequences. It is important to maintain an appropriate size of the threads and secure strong support from the administration for adequate resources to make the threads model successful. Many computing programs are seeking accreditation or reaffirming their accreditation to ensure that education provided by them meets acceptable levels of quality. Under the threads model, the combination of any two threads comes up with a degree. This means that we have to design our threads in such a way that the union of any two threads must satisfy the accreditation requirements for a computing degree. In terms of thread outcomes, this means that we have to design our threads such that all general program outcomes (a) – (i) listed in [10] must be satisfied as well as program specific outcomes for individual computing programs. Threads are not tracks or concentrations. Then the questions we are facing will be: What is the relationship between a thread and a concentration? A minor?

ACKNOWLEDGMENTS
The research work was supported by the NSF CPATH grant #0722157.

REFERENCES