Response to Tidal Wave II

Phase II: New Programs

Information Technology Fluency for Undergraduates

Marti Hearst, Assistant Professor
David Messerschmitt, Acting Dean
School of Information Management and Systems

Shankar Sastry, Chair
Christos Papadimitriou, Associate Chair
Jan Rabaey, Associate Chair
Department of Electrical Engineering and Computer Sciences

Oct. 28, 2001
Information Technology Fluency for Undergraduates

Executive Summary

We are alerting the campus to a problem that is evident today and likely to become much worse in the future; that is, the need for many undergraduates across campus to obtain an understanding of and comfort with computing and information technology concepts and skills, and the current inability of UC Berkeley to accommodate this need. A “Task Force on Computer Science Undergraduate Education” in the EECS department last year identified this as an important unfulfilled need, but also determined that the EECS department is, due to its oversubscription in its current mission, unable to take on this burden. Two solutions are to distribute courses in information technology throughout different academic units in the university, as is happening by default today, or create a common curriculum for the campus that can serve as a foundation for discipline- or field-specific advanced courses. We propose the latter, and outline a suggested curriculum and propose mechanisms for administration of this program.

Our Proposal

Information technology in all its manifestations is critical to the personal and professional lives of our students. The National Research Council Report1 “Being Fluent with Information Technology” eloquently makes the case that computer "literacy" -- as traditionally understood on this campus the ability to use existing technology -- sets the bar too low. The NRC argues that the tendency for technology to change rapidly, paired with its importance to almost every field, implies the need for most undergraduate students to acquire an understanding of the underlying fundamentals of information technology, and to be able to apply it in new circumstances. They call this competency Information Technology (IT) Fluency. Today, IT Fluency is probably as important as fluency in writing, and many campuses have adopted some form of IT fluency requirement.

Unfortunately, at Berkeley we currently do not offer our undergraduates even the opportunity to gain IT Fluency. One current offering for non-technical majors is IDS 110, an enormously popular introductory programming course2. Although IDS 100 is valuable, it is only an introduction and is focussed on the specific topic of Web interface programming. The introductory courses for computer science majors are seriously over-subscribed and their content is too tailored to majors to be useful to a broader crosssection of students3.

---

1 Avilable online at http://www.nap.edu/catalog/6482.htm

2 Of the two programming courses taught by EECS, CS 9 series assumes the students already know how to program and CS 3 teaches symbolic programming, a course which is not very useful for non-majors.

3 The lower-division computer science courses CS 61A, 61B, and 61C are oriented towards those students intending to major in computer science. The first course, CS 61A, is difficult, competitive, time consuming, and tailored to the student body who continue in the computer science program. Furthermore, because it is part of a three-semester series geared towards majors, a non-technical student who completes this course alone does not learn many of the most important concepts, which appear in later courses. Thus
An increasing number of non-technical majors require some form of information technology abilities and understanding. These include the areas of media studies, journalism, business, art, music, architecture, communication, the biological sciences, cognitive science, and others. One way to address this is for each such program to develop its own information technology courses, as for example the current Business 147/148 courses on e-commerce. While such courses have field-specific content, they are also forced to cover many fundamentals, and to do so in a non-systematic and haphazard manner. The proliferation of such courses would result in a hodgepodge of partially duplicative courses.

Another approach is to create a common curriculum in IT fluency, and then allow individual fields to build on this with supplementary courses specific to their needs. That is the approach we recommend, because then the fluency courses can be systematic, defined and taught by experts, and taught efficiently. We propose that the campus offer a carefully designed set of courses, which can be taken by non-technical majors either as individual courses or as part of a minor or concentration in information technology. These courses would have as a primary goal equipping students with the ability to conceptualize the effective use of IT in their area of concentration, and would not be designed to be a major or specialty in their own right -- students would be better off majoring in computer science if this is their goal. To illustrate the scope and philosophy of this vision, a preliminary curriculum is displayed in the appendix.

We expect that IT fluency courses would be extremely popular, with large enrollments. They would also be ideal courses for summer teaching, since they would merely supplement some primary course of study. How would we accommodate this large teaching load? Although the Computer Science Division would be the most obvious home for such an effort, the current extremely heavy teaching loads makes CS unable to take this on. The School of Information Management and Systems (SIMS) might be another home, but is far too small to assume this responsibility. Furthermore, SIMS has a much broader interdisciplinary mission, and hence only a few faculty who are able to teach the technical courses. Increasing the size of this faculty commensurate with the needs of an IT fluency undergraduate program would result in a substantial imbalance in its faculty in relation to its research and graduate teaching mission.

We thus advocate an alternative mechanism which was first proposed last year by the aforementioned EECS Task Force. This mechanism has two parts. First, the Berkeley Division of the Academic Senate would establish a new Committee on Education in Information Technology, which would take responsibility for coordinating the offerings of various departments relating to IT with the centralized offerings. Secondly, the centralized courses and their minor would have a home in a new "Information Technology Studies (ITS) Program". ITS would be housed in the College of Engineering, SIMS, or another school or college, similar to the current Interdisciplinary Studies Program in the College of Engineering. ITS would be staffed by an academic administrator and other permanent staff. It would not have ladder faculty assigned to it, nor do research or grant degrees. Its role would be limited to the development and offering of IT

---

61A does not teach the appropriate topics for non-majors, and, worst of all, would scare many students away from information technology topics altogether.

4 Cornell University recently reorganized a significant portion of their campus to address precisely these issues, deciding on just such a centralized solution; see http://www.cs.cornell.edu/current_news/Report_of_the_Task_Force.htm

5 As evidence, the current IDS 110 has an enrollment of about 500 students per semester.
fluency courses, working under the governance of the Academic Senate. ITS could draw upon existing academic units for instructors of such courses (we expect there to be an increasing number of faculty in various units who themselves possess technical abilities), or hire permanent or outside instructors (especially in the summer). The entire budget of the IT fluency program would reside in ITS, and it would either reimburse existing units to teach its courses, or those units could be ‘paid’ from a pool of permanent faculty slots specifically allocated to the classroom teaching needs of ITS.

We believe that EECS and SIMS should provide strong intellectual guidance for the ITS program, both initially and on an ongoing basis. For this purpose, we suggest that ITS have a governance board comprising faculty drawn from these two units as well as other academic units who are primarily ‘consumers’ of these courses.

Appendix: Suggested IT Fluency courses

The following is an illustrative and preliminary list of suggested IT fluency courses, together constituting a systematic introduction to information technology and its applications (see the NRC Report “Being Fluent in IT” for a more comprehensive syllabus for such a curriculum). Three kinds of courses are proposed:

- A broad introductory survey course (ITS 1) that considers information technology in a social context. A student taking this course alone would have an introduction and broad overview, and be in a position to track future developments in computing. This is also an attractor course, motivating students to want to learn more and perhaps minor in ITS or major in CS (the latter could be a significant opportunity to increase the diversity of CS majors). The remaining courses would use ITS 1 as a prerequisite, and thus could assume contextual knowledge.

- Foundational technical courses, including programming, targeted at novice programmers. ITS 3 is the centerpiece giving an overview of fundamental concepts and skills of computer science. After taking ITS 2 and 3, students should be qualified to take certain specific CS upper-division courses such as CS 160 (User Interface Design) and CS 169 (Software Engineering).

- Courses that place information technology in an application and societal context, relating the technology to its use and giving students practice and skills in applying information technology in new ways.

ITS 1: Introduction to Information Technology and its Applications. Overview of information technology, including processing, storage, and communications. Centralized and distributed computing systems, software creation, systems integration, operations. Major applications of information technology and its role in society. Hands-on experience with rapid prototyping tools and conceptualizing new applications.

ITS 2a, 2b, 2c: Introductory Programming for Novices. Similar in methodology to introductory programming courses such as the CS 9 series, but oriented towards students with little or no experience programming in a high-level language. Prerequisite ITS 1.

ITS 3: Foundations of Software Design. Introduction to computing foundations. How computers work, from bits and binary to CPUs and operating systems; analysis and choice of
standard algorithms and data structures; introduction to formal languages, including regular expressions and context-free grammars; introduction to computability and Turing machines. Principles of object-oriented design and UML, hands-on experience in object-oriented programming. Prerequisite ITS 1 and ITS 2a-c.


ITS 11: Programming and Engineering the Web. Technological and engineering foundations of the Web. HTTP protocol and server engine, technologies and programming languages used to develop Web applications (e.g., content, e-commerce and portal sites and services, etc.). Quantitative methods for performance evaluation, scalability, and capacity planning. Hands-on programming experience, semester-long group project that designs, programs, deploys and evaluate a fully functional Web-based application. Experience in use of open-source technologies (e.g., apache, mySQL, linux, perl). Prerequisite ITS 1 and ITS 2a or ITS 10.

ITS 14: Networking and its Applications. Types of computer applications that emphasize networking and supporting technologies, such as collaboration, coordination, communities, information access, enterprise, and commerce. Network protocols and performance, middleware, particular issues in distributed computing, security protocols, layering architectures. Prerequisite ITS 1.

ITS 16: Information Architecture. Technology and information architectures for distributed computing applications, and its relation to organizational design. Experience in architectural design in support of organizational missions. Information capture, management, organization, and access. Experience in programming and deploying a distributed application. Prerequisite ITS 1.

ITS 20: Business and Economics of Computing. Influence of the market on the diffusion of new computing and communications technologies. Includes standardization, managing portfolios of intellectual property, government policies, industry structure, management challenges, information economics. Major group project experience in analyzing a significant case. Prerequisite ITS 1 and ITS 2a or ITS 10.

ITS 40: User Interfaces for Computing. The design, implementation, and evaluation of human/computer interfaces. Interface devices (keyboard, pointing, display, audio, etc.), metaphors (desktop, notecards, rooms, ledger sheets, tables, etc.), interaction styles and dialog models, design examples, and user-centered design and task analysis. Interface-development methodologies, implementation tools, testing, and quality assessment. Students will develop a direct-manipulation interface. Prerequisite ITS 1 and 2a-c. (Note: CS 160 might be a substitute for this course.)

ITS 50: Introduction to Software Engineering. Social and management considerations surrounding large-scale software development. Methodologies and experience in working with end-users to define the requirements, software architecture, and development. Comparative analysis of alternative processes, such as waterfall, iterative, spiral, and community-based. Prerequisite ITS 1 and ITS 2a or ITS 10. (Note: CS 169 might be a substitute for this course.)