Designing an IT College

Peter J. Denning, Ravi Athale, Nada Dabbagh, Daniel Menascé, Jeff Offutt,
Mark Pullen, Steve Ruth, and Ravi Sandhu
George Mason University, Fairfax, VA 22030 USA
{pjd | rathale | ndabbagh | menasce | ofut | mpullen | ruth | sandhu} @ gmu.edu

Key words: information technology school, core curriculum, progressive competence, assessment, professional education

Abstract: The University of the United Arab Emirates (UAEU) commissioned an international panel of experts to devise a model curriculum for their new College of Information Technology. The model has a university core in the freshman year, an IT core in the second year, and a professional concentration in one of seven degree programs in the third and fourth year. The seven degrees are computer science, computer systems engineering, software engineering, information systems, network engineering, information security, e-commerce, and educational technology. The curriculum is structured around a progressive-competence model, whose graduates meet recognised standards for entry-level professionals.

1. THE GROWING DEMAND FOR IT COLLEGES

The first academic departments of computing, computer science, computer engineering, and infomatics were organised in the 1960s. Their number has multiplied world-wide: more than 180 PhD-granting departments exist in North America alone. Departments offering Information Systems degrees have multiplied likewise. Software Engineering shows every sign of following the same path. In the 1990s Information Technology (IT) came to pervade the university: many departments are setting up IT groups, requiring IT fluency of all undergraduate students (NRC 2000), and encouraging students to take IT minors. The ACM and IEEE Computer Society have completed work on a new core curriculum for computer science and engineering, emphasising the need for the core to serve CS and other IT disciplines as well (2001), a departure from the tradition of a core for CS&E alone (ACM 1991, 1997; Denning *et al*. 1989). Meanwhile, a large network of over 1500 corporate universities offers on-the-job IT training for hundreds of thousands of people each year (Corporate University 2001).

Computing is no longer simply a discipline, but is part of a large emerging profession with already over three dozen organised specialities (Denning 1998, 2001a, 2001b). Education of IT professionals can no longer be the responsibility of a single university department or degree program. IT curricula must include a professional body of knowledge complementing the intellectual body of knowledge. Many universities are seriously investigating whether and how to consolidate several computing-related departments or degree programs within a single academic unit. The design of curricula for these new schools is a growing concern (Tsichritzis 1999).

The pioneers of IT colleges in the USA are the School of Information Technology and Engineering at George Mason University (1985), the School of Computer Science at Carnegie-Mellon University (1988), the College of Computing at Georgia Tech (1991), the School of Information Science and Technology at University of Nebraska Omaha (1996), and the School of Informatics at Indiana University (2000). The University of Nebraska Omaha published a curriculum for an IT school (Lidtke *et al*., 2000). Tony Greening assembled a collection of proposals for IT curricula (Greening 2000). In Fall 1999, the Chancellor of the University of the United Arab Emirates commissioned a panel of experts well versed in the above trends to design a model curriculum for their newly formed IT College. The remainder of this paper will present the model and discuss its innovations. The model is designed to help any other university organise a coherent IT curriculum in a single academic unit. A full copy of the panel report is on a website (Denning 2000).

1.1 The IT Field

The IT field has been maturing rapidly from a set of narrowly focused, autonomous disciplines into a broad profession (Denning 1998, 2001a). The IT profession comprises all the professionals who make a living helping other people with their IT concerns, breakdowns, and opportunities. The forty or so concentrations of the profession fall into three groups: the *IT specialities* are concerned with information technology itself; the *IT-intensive disciplines* pursue other concerns with a heavy dependence and investment in IT; and the *IT infrastructure* areas are professionals who implement, maintain, operate, and repair IT.

Table 1. The Information Technology Field

IT Specialities	IT-Intensive Disciplines	IT Infrastructure
Artificial intelligence	Banking and Financial	Computer technician
Computer science	Bioinformatics	Help desk technician
Computer engineering	Cognitive science	Network technician
Computational science	Digital library science	Professional IT trainer
Database engineering	E-commerce	Security Specialist
Graphics	Genetic engineering	System administrator
Human computer interaction	Information science	Web services designer
Network engineering	Information systems	Web identity designer
Operating systems	InfoSec and Privacy	
Performance engineering	Instructional design	
Robotics	Knowledge engineering	
Scientific computing	Mgt information systems	
Software architecture	Multimedia design	
Software engineering	Telecommunications	
System security	Transportation	

The IT field is thus much more complex than computer science and presents major new challenges to educators. A single university department cannot serve it. The smallest organisational unit that can is a school. But an IT school cannot be neatly segregated into autonomous departments like many other schools because the professional areas share significant bodies of knowledge. The degree programs offered by the school are best designed through a co-ordinated effort.

Because the IT field is dominated by professional specialities, not just intellectual disciplines, its students need to be immersed in practice as much as in a study of concepts and principles. They need to learn to be professionals, which means they need to learn and embody their speciality's body of knowledge, its standards of practice, and its ethical codes of conduct.

Designing a curriculum for a set of related IT professional degrees is not the only challenge. Because the half-life of technical knowledge is so short -- and with "Internet time" continues to get shorter -- an IT school must pay special attention to fostering in its graduates the ability to continue learning both new technologies *and new practices* throughout their careers. The curriculum must present students with many opportunities to practice mobilising their resources and creativity -- including negatives such as incompletely specified projects, fickle and demanding customers for project teams, and midstream changes of direction. Students need to develop a tolerance for uncertainty and a facility to recover from breakdowns.

In addition to being fast-paced, the IT market features a model of innovation that is quite different from the research model cherished in universities (Tsichritzis 1997). The market model emphases fast time-to-market; it funds projects with venture capital; it fosters innovation by rewarding entrepreneurs who specialise in transforming people's practices to become more productive with the help of new technology. The university research model emphasises careful development of ideas and principles; it funds projects by federal and state grants; it achieves innovation only after ideas have gone through a long "pipeline" that distils out the best and brings them into practice (over 15-20 years). The dichotomy between market and idea-pipeline models presents great challenges to educators. Graduates of IT programs need to understand the market model as well as the idea-pipeline model. Most faculty have not experienced the market model first hand. Intensive market involvement of undergraduates can be accomplished by involving them in industry through internship programs and by offering training in the skills of entrepreneurship.

In summary, the design of an IT curriculum is complicated by factors in the IT market and by the complex structure of the IT profession.

2. A MODEL CURRICULUM

2.1 Degree Programs

Our model for the IT school curriculum encompasses seven degree programs and one certificate program, all based on a common IT core curriculum. The *Computer Science* (CS) Curriculum is consistent with the principles of the new ACM/IEEE Computing Curriculum (2001). The *Information Systems* (IS) Curriculum is based on the ACM IS'97 model with updates (1997). The *Computer Systems Engineering* (CSE) Curriculum is a variation of Computer Engineering, embodying a modern mix of signals, circuits, and software. The other four degree programs -- *Software Engineering* (SWE), *Network Engineering* (NE), *Information Security* (SEC), and *Electronic Commerce* (EC) -- are the first (or are among a very few) rigorously technical undergraduate programs anywhere in the world. The *Educational Technology* (ET) certificate can be offered to students outside the IT school. The curricula for these programs, along with a catalogue of their 135 courses, a matrix representation of the IT body of knowledge, and an IT core curriculum can be found on report website (Denning 2000)

2.2 The Body of Knowledge

The body of knowledge consists of two parts. The *intellectual body* sets forth the knowledge areas with subdivisions into theory, abstraction, design, and technology. The *professional practices body* sets forth levels of competence that exist in every speciality.

Intellectual Body. Following the model of the ACM/IEEE-CS report, *Computing as a Discipline* (Denning 1989), we specified a body of intellectual IT knowledge as a matrix with rows corresponding to knowledge areas and columns for theory, abstraction, design, and technology. (The technology column is new.) Whereas the computing body of knowledge contained 10 knowledge areas in 1989, the IT body contains 48 areas today. Examples of IT knowledge areas are architecture, digital devices, software design, network protocols, cryptographic protocols, business systems and processes, management information systems, and instructional design. The granularity of these areas is just broad enough to correspond to professional specialities that have their own identity, recognised literature, and communities of practice. Each area hosts one or more courses in the curriculum. Except for survey courses, no course cuts through multiple areas.

Professional Body. Professionals progress in competence over time through six levels whose standards of performance are set by the community:

Beginner
Advanced beginner
Entry-level professional
Proficient professional
Expert
Virtuoso
Master

These are levels of *embodied competence* -- i.e., performance in action. It takes time, practice, and experience for a person to attain a level. One's professional career can be interpreted as a journey on a path to mastery.

Our model delivers entry-level professional competence, a challenge not often met by IT undergraduate programs.

Students at the *Beginner Level* (sophomore year) are aware of the structure of the IT field and the nature of the work in the various specialities; they able to develop algorithms, data structures, and simple circuits to solve well-defined problems; they are able to program and test their algorithms and circuits. While not expected to see the connections and interrelations among the components of typical computing systems, they are expected to understand the purpose of each component.

Students are the *Advanced Beginner Level* (junior year) are familiar with the terminology and concepts of the speciality; they see many connections among components of computing systems; they are able to design algorithms (and possibly circuits) of moderate complexity (several dozen modules), program them, and test them. They are able to carry out tasks for a customer but need supervision to avoid common pitfalls and breakdowns. They are able to communicate effectively in speaking and in writing.

Students at the *Entry-level Professional Level* (senior year) are thoroughly familiar with the speciality; they understand systems and can diagnose system problems; they are able to design systems of moderately large complexity (hundreds of modules), program them, test them, document them, and present them. They can carry out standard professional tasks for customers in application domains without supervision. They practice professional ethics.

2.3 Exhibitions and Their Management

Our model is designed to lead students to the entry-level professional level at graduation. Each year concludes with an *exhibition*, a major milestone at which students perform tasks demonstrating knowledge. Not only are exhibitions a valuable experience for students, they are the primary assessment tool for measuring whether a student has attained a level of competence.

We do not recommend using exhibitions as gateways between stages. That would create a significant management overhead to provide non-passing students with additional practice and an opportunity to retake the exhibition. We do recommend that faculty advisors work closely with students who do not pass exhibitions to help them prepare for the courses in the next year of study.

Exhibitions are an uncommon practice within a university. Their management challenges should not be a reason to abandon them. They program must be carefully planned and executed.

Exhibitions can be packaged as regular courses. Each section can contain 30 students, organised by the instructor into 3-person teams. The instructor's time will be devoted primarily to managing the teams through weekly progress reports and individual team meetings. A few all-hands meetings will be required; in them, the instructor can review common rules, constraints, and knowledge, and have the students perform dry runs of their project presentations. At the end of the exhibition course, all the teams will present their projects in a final exposition. An Exhibition Week can be reserved late in the spring semester to enable all students and faculty to attend exhibitions and to celebrate the completion of the academic year.

Staff support will be needed to help link student teams with industry customers, place the students into sections according to common themes, obtain the meeting and exhibition space, and otherwise to manage the exhibition program.

2.4 The IT Core

The UAEU has a standard freshman year for all students covering 26 hours of language, culture, humanities, mathematics, and technology. The professional degrees each need a total of 132 credit hours for graduation. Of the remaining 106 credit hours, we allocated 42 for the IT core (sophomore year including summer) and 33 hours each for the junior and senior years including their exhibitions.

One of the novel aspects of our model is its IT core curriculum. The core consists of 42 credit hours concentrated mostly in the sophomore year as shown in Table 2. IT majors substitute the first course in programming for the freshman "computer literacy" course. Except for the Professional Responsibility Workshops, all courses are 3 credit hours.

Instead of one 3-credit ethics and professionalism course that will soon be forgotten, we recommend three one-credit professional responsibility workshops conducted 1 hour per week for one semester in each of the sophomore, junior and senior years. We also recommend a course on speaking and writing in the junior year; this course can be taught by a humanities department after the IT College has approved its content. A more advanced programming course, not part of core, is available for some of the majors.

In addition to the 132 credit hours to attain a professional IT degree, we recommend that each student be required to participate in an industry internship for one semester (15 credit hours). This work is best done in one or more of the summers.

Our full report addresses operational issues that must be taken care of if the IT College is to succeed. These include: managing exhibitions schoolwide, course websites as part of the corporate memory of the curriculum, recruitment of good faculty, training of faculty, facilities management, distance instruction, team teaching, and accreditation. We recommend that the dean appoint a chief academic officer for the school to co-ordinate these aspects and maintain the curriculum's quality, currency, and integrity.

Table 2. Overview of the IT Core Curriculum

AREA	F	SOPHOMORE			J & S
Math		Calculus concepts (derivative, integral, simple diff eq, linear algebra)	Probability and statistics	Discrete math	
Systems		Digital hardware and communication	Computing Systems (OS, networks, architecture)	Information Systems (Web, database, security)	
Programming	Prog I	Prog II			(Prog III)
Business			Business basics	Enterprise basics	
Science			Biology Concepts	Physics Concepts	
Communication					Speaking and writing
Professional responsibility		Sophomore Professional responsibility workshop			J & S Professional responsibility workshops
Exhibition				Core exhibition	

3. SUMMARY

Our curriculum recommendation is attuned to the structure of the emerging IT profession and is designed to help graduates be effective as IT professionals. The model has a common IT core to be used in the second (sophomore) year by all majors, followed by two years in the respective majors. Students progress through the curriculum by a *progressive-competence model*, not simply by accumulating course hours. At the end of each year, students demonstrate their knowledge through an *exhibition*. The three exhibitions represent three major milestones, certifying that students have attained beginner, then advanced beginner, and finally entry-level professional competence in their chosen degree field. Within this structure, we have offered a rigorous curriculum that emphasises math, science, humanities, and technology in each of the degree programs.

In addition to the exhibitions, the model contains several innovations. The IT Core (sophomore year) is shared among all the degree programs of the College. Three programs -- E-Commerce, Network Engineering, and Information Security -- may be the first-ever rigorously technical undergraduate programs. The Software Engineering program is one of a few specified for undergraduates. The Educational Technologies certificate may be the first program of its kind outside a college of education that enables technologists to assist other educators design and deliver high quality educational materials. A strong theme of business and entrepreneurship pervades the curriculum, starting with the core business requirements and including a semester of industry experience. There is also a strong emphasis on systems thinking through the curriculum -- specification, integration, measurement, testing, modeling, and evaluation.

REFERENCES

ACM (1991). *Curriculum 91, Recommendations for undergraduate computer science degree programs*. ACM Inc., 1515 Broadway, 17th Floor, NY, NY 10036. [www.acm.org] 5.1.2001

ACM (1997). *Curriculum IS'97: Recommendations for undergraduate information systems degree programs.* ACM Inc., 1515 Broadway, 17th Floor, NY, NY 10036. [www.acm.org] 5.1.2001.

ACM & IEEE-CS (2001). Computing Curriculum 2001: Draft recommendations for new undergraduate computer science and engineering degree programs. [www.acm.org/sigcse/cc2001] 5.1.2001.

Corporate University Xchange (2000). An organization of 1500 corporate universities. [www.corpu.com] 5.1.2001.

Denning, P., D. Comer, D. Gries, M. Mulder, A. Tucker, J. Turner, P. Young (1989). Computing as a discipline. *Communications of ACM 32*, 1, pp. 9-23. A condensed version was published in *IEEE Computer*, February 1989.

Denning, P (1998). Computing the profession. *Educom Review 33* (Nov-Dec), pp. 26-30 & 46-59.

Denning, P., R. Athale, N. Dabbagh, D. Menascé, Jeff Offutt, Mark Pullen, Steve Ruth, and Ravi Sandhu (2000). A Model Curriculum for the United Arab Emirates University. [cne.gmu.edu/pjd/UAE] 5.1.2001.

Denning, P. (2001a). Who Are We? *ACM Communications 44*, 2 (February), pp. 15-19.

Denning, P. (2001b). Crossing the Chasm. *ACM Communications 44*, 4 (April), pp. 21-25.

Lidtke, D., G. Stokes, and J. Haines (1999). *ISCC'99: Information Science-Centric Curriculum*. See [www.iscc.unomaha.edu/] 5.1.2001.

Greening, T. (2000). *Computer Science Education in the 21st Century.* Springer-Verlag.

National Research Council (2000). *Becoming Fluent in Information Technology.* A panel report from the Computer Science and Technology Board. See [books.nap.edu/catalog/6482.html] 5.1.2001.

Tsichritzis, D. (1999). Reengineering the university. *Communications of ACM 42*, 6, pp. 93-100.

Tsichritzis, D. (1997). The Dynamics of Innovation. In *Beyond Calculation: The Next Fifty Years of Computing* (P. Denning, Ed.). Copernicus Books.

BIOGRAPHIES

Peter J. Denning is a professor of computer science at George Mason University. He is noted for leadership in computer science education, where he successfully advocated that operating systems be part of the core curriculum (1972), organised a new framework for the computing core curriculum (1989), and recognised information technology as a profession (1998). He was a pioneer in the early development of operating systems, contributing a solid scientific case for virtual memory, the working set model for program behaviour, and computer architectures that support OS functions. He was a co-founder of CSNET, the community network that bridged between the ARPANET and modern Internet. He was founding director of the

Research Institute for Advanced Computer Science (RIACS) at NASA Ames Research Center. He was President of ACM (1980-82) and is currently chair of the ACM Education Board. He has published over 280 scientific articles and 6 books. He holds two honorary degrees, two teaching awards, three distinguished service awards, three professional society fellowships, two best-paper awards, and the prestigious ACM Karlstrom Educator Award.

Ravi Athale received his bachelors and masters degrees Physics in India and his PhD (1980) in Electrical Engineering from University of California San Diego. At the US Naval Research Laboratory, in Washington, DC, and later at the BDM Corporation he conducted research on optical signal and image processing and optical computing. He is Fellow of the Optical Society of America and a member of the IEEE Lasers and Electro-Optics Society. He joined George Mason University as Associate Professor in the Electrical and Computer Engineering Department (1990). He was a joint developer of an information-technology oriented freshman introductory course for Electrical Engineering students.

Nada Dabbagh is an Assistant professor in the Instructional Technology program at George Mason University. She teaches graduate courses in Instructional Design, Web-Based Instruction, Applied Psychology, and Technology Integration. Her main research interests are the design and evaluation of Web-Based Instruction (WBI), problem generation and representation in Problem-Based Learning (PBL), and the contextualizing of instruction through Constructivist Learning Environments (CLE). She holds a doctorate in Instructional Systems Design (INSYS) from The Pennsylvania State University (1996). She has been studying the pedagogical effectiveness of traditional courses that have been redesigned for online delivery using Web-based Course Authoring tools.

Daniel A. Menascé is a Professor of Computer Science and the co-director of the E-Center of Excellence in Research and Education in E-Business at George Mason University. He received a PhD in Computer Science from UCLA (1978). He has published over 110 technical papers in computer systems and performance evaluation, and is chief author of five books on these subjects. He is a Fellow of the Association for Computing Machinery (ACM). He received two best-paper awards for his research. He received two teaching awards at George Mason University. He is a frequent consultant in performance modeling, capacity planning of e-commerce network environments, and software performance engineering. He is a popular tutorial teacher at major research conferences.

A. Jefferson Offutt is an Associate Professor in the Department of Information and Software Engineering at George Mason University. He teaches MS and PhD courses in Software Engineering and has developed new courses in software testing, construction, design, user interface design, experimentation, and analysis. His current research interests include program testing, automatic test data generation, object-oriented program analysis, module and integration testing, software architecture-based system testing, formal methods, and change-impact analysis. He has published over 50 research papers. He holds a PhD from the Georgia Institute of Technology (1988).

J. Mark Pullen is an Associate Professor of Computer Science and a member of the C3I Center at George Mason University, where he heads the Networking and Simulation Laboratory. He holds BSEE and MSEE degrees from West Virginia University, and the ScD in Computer Science from the George Washington University. He is a licensed Professional Engineer and a

Fellow of the IEEE. Prior to joining the GMU faculty he was an officer in the US Army, in which capacity he served four years on the faculty of the US Military Academy at West Point, New York and seven years at the Defense Advanced Research Projects Agency (DARPA). At DARPA he managed programs in high performance computing, networking, and simulation. He teaches courses in computer networking, and has active research projects in networking for distributed virtual simulation and networked multimedia tools for distributed education. He received the IEEE's Harry Diamond Memorial Award for his work in networking for distributed simulation.

Stephen R. Ruth is Professor of Technology Management in the School of Public Policy at George Mason University, and Director of the International Center for Applied Studies in Information Technology (ICASIT). His research concerns strategic planning associated with leveraging the use of information technology in large organizations, with particular emphasis on the effect of Knowledge Management policies on the work of dispersed teams. He has consulted widely throughout the world. He received his BS from the US Naval Academy and MS from the Navy Postgraduate School, and served 23 years in the Navy, retiring with the rank of Captain. His PhD is from the Wharton School, University of Pennsylvania. He is author or co-author of over 100 published articles and 4 books.

Ravi Sandhu is Professor of Information and Software Engineering and Director of the Laboratory for Information Security Technology at George Mason University. He holds PhD and MS degrees from Rutgers University, and BTech and MTech degrees from IIT Bombay and Delhi respectively. His principal research and teaching interests are in Information and Systems Security where he is widely known for his expertise in role-based access control and database security. He has published over 130 technical papers on computer security. He is the founding editor-in-chief of the ACM Transactions on Information and Systems Security (TISSEC), and is security editor for IEEE Internet Computing. He founded the ACM Conference on Computer and Communications Security and the ACM Symposium on Access Control Models and Technologies, and continues to chair their steering committees. He is currently Chairman of ACM's Special Interest Group on Security Audit and Control (SIGSAC).