Journal of Information Technology Education Volume 5, 2006

The Information Technology Model Curriculum

Joseph J. Ekstrom,
Brigham Young University, Provo, Utah, USA

jekstrom@byu.edu

Sandra Gorka,
Pennsylvania College of Technology, Williamsport, PA, USA

sgorka@pct.edu

Reza Kamali,
Purdue University Calumet, Hammond, IN, USA

kamalir@calumet.purdue.edu

Eydie Lawson,
Rochester Institute of Technology, Rochester, New York, USA

eal@it.rit.edu

Barry Lunt,
Brigham Young University, Provo, Utah, USA

luntb@byu.edu

Jacob Miller,
Pennsylvania College of Technology, Williamsport, PA, USA

jmiller3@pct.edu

Han Reichgelt,
Georgia Southern University; Statesboro, Georgia, USA

han@georgiasouthern.edu

Abstract

The last twenty years has seen the development of demand for a new type of computing professional, which has resulted in the emergence of the academic discipline of Information Technology (IT). Numerous colleges and universities across the country and abroad have responded by developing programs without the advantage of an existing model for guidance. Efforts to define a model curriculum for IT began at the first Conference on Information Technology curriculum (CITC-1) in December 2001, which included representatives from 15 IT programs at four-year universities across the United States. Membership in SIGITE (Special Interest Group Information Technology Education) has grown to over 400 representing many of the four-year IT programs in the United States and abroad and some of the two-year IT programs in the United States. Continued development of the curriculum and subsequent funding by the Education Board of ACM enabled the completion of a first draft of the model curriculum for IT establishing program outcomes and a body of knowledge defining the discipline. This paper presents an overview of the process followed and the results achieved.

Keywords: Information Technology, curriculum, information systems, computer science.

Editor: Mike Hart

Background

The last 20 years or so has seen a dramatic growth in the demand for professionals trained in information and communication technologies. This growth was partly driven by tremendous technological advances, such as the emergence of computer networking, including wireless networking, graphical user interfaces, and the Internet, the World-Wide Web, and their applications. But the growth in demand was also driven by a much greater realization on the part of many organizations and individuals of the importance of information and communication technologies for their well being, and the more widespread use of these technologies by individuals whose primary area of expertise was not in information technology (Dalhom & Mathiassen, 1977; Denning, 2001a; Denning 2001b).

While programs in traditional computing disciplines such as computer engineering and computer science responded to these new trends, albeit often slowly, employers started to demand skills that graduates typically did not get out of a traditional computer science or computer engineering program. For example, many traditional computer science programs did not equip their graduates with the practical network or system administration skills that organizations needed to expand and maintain their IT infrastructures, or the web development skills required to take advantage of the many opportunities opened up by the Internet. Moreover, because of the more theoretical emphasis of computer science programs, graduates often did not acquire a sufficient understanding of organizational processes to be able to support IT applications from a user or organizational perspective (Denning, 2004).

While programs in Information Systems are able to offer material to their students that allow them to develop a better understanding of organizations and the way in which IT applications can support them, they often do so at the expense of a sufficiently thorough coverage of the technology. This has been exacerbated by the fact that a good number of programs in Information Systems were offered in business schools and were limited in the number of courses they could offer in IT because of accreditation standards for business schools in place at the time.

It was in this environment that a number of universities started to offer undergraduate programs in IT. While there was, and still is, far less homogeneity among undergraduate programs in IT than there is among programs in computer science or information systems, all IT programs show a “family resemblance” to each other. Most cover areas such as networking, web development and system administration in some detail, while very few pay particular attention to the theoretical foundations in complexity theory that is so prevalent in computer science programs. Until fairly recently, most IT programs arose in isolation from each other, often evolving out of existing programs in computer science, computer engineering or computer engineering technology, or information systems (Lunt et al., 2003a; Reichgelt et al., 2004). However, there was also a growing realization on the part of those programs that were aware of each other’s existence that a forum was needed in which they could discuss the various features of their programs, and improve their programs based on the experiences of IT programs at other institutions (Lunt et al., 2003b).

Thus, in early 2001, a steering committee composed of five universities offering or getting ready to offer IT programs compiled a list of IT programs across the United States with the intention of organizing an invitation-only conference. These efforts came to fruition in December 2001, when representatives from 15 colleges/universities attended the first Conference for IT Curriculum (CITC-1) in Provo, Utah. The primary aims of this conference were to establish a national organization of IT educators and begin to establish academic standards for this rapidly emerging discipline (Lunt et al., 2004, 2005).

During CITC-1, the Society for Information Technology Education (SITE) was established. Late in 2002, SITE morphed in to an ACM Special Interest Group on IT Education (SIGITE). CITC-1 also saw the establishment of two committees, one to formulate accreditation standards for programs in IT and one to formulate a model curriculum for IT. Finally, at CITC-1, a Delphi study was conducted to determine which topics participants thought should be covered in any IT program (Lunt et al., 2004). Work on both the curriculum and the accreditation criteria continued at subsequent conferences, namely CITC-2 (April 2002), CITC-3 (September 2002), and CITC-4 (October 2003).

SIGITE approved a set of accreditation criteria at CITC-4 and forwarded these to ABET, the main US accreditation agency for programs in computing, engineering, engineering technology, and the applied sciences, formerly known as the Accreditation Board for Engineering and Technology. ABET works through four commissions, each having responsibility for accrediting programs in similar areas. For the purposes of this paper, the most important commissions are the Engineering Accreditation Commission (EAC), which is responsible for accrediting programs in engineering, including computer and software engineering, and the Computing Accreditation Commission, which is responsible for accrediting programs in Computer Science, Information Technology and Information Systems. After some reformulation and reorganization, these criteria have been incorporated in the recently approved ABET CAC accreditation criteria (ABET CAC, 2005)

In parallel to the efforts to formulate accreditation criteria for programs in IT, SIGITE also continued its work on formulating an IT model curriculum, both at its conferences, and on a continuing basis through its curriculum committee. In October of 2004 funding was obtained from the Education Board of ACM to support completion of the model curriculum effort, and in April 2005, an initial version of the IT model curriculum was released for public comment, some of which was incorporated into a second release of the document in October 2005 (SIGITE Curriculum Committee, 2005)

The intent of this paper is to report on both the process by which the model curriculum was formulated and on the current draft curriculum itself.

Process

As said, the effort to write the IT model curriculum started at CITC-1 with the establishment of an IT model curriculum committee. Initially, the committee consisted of around 15 individuals and grew fairly rapidly to a group of around 25 IT educators. It became clear that it would not be feasible for a group of this size to write an IT model curriculum, and so a writing committee was appointed consisting of the authors of this paper. Although the writing committee was responsible for actually producing the various drafts of the IT model curriculum, it solicited and received input from the wider curriculum committee. In addition to reviewing the various drafts produced by the writing committee, many of the members of the curriculum committee were responsible for significant parts of the curriculum document.

In this section, we discuss some of the principles that the curriculum committee established in order to guide its deliberations, the inputs that it used in formulating the curriculum document, some of the decisions that it made about the structure of the curriculum document, and the detailed processes that were followed to ensure that there was a relatively broad participation in the curriculum writing process.

Principles

Early on in the curriculum development process, the IT model curriculum committee established a number of principles that it would use in formulating its curriculum recommendation. The most significant of these were:

The formulation of the curriculum was governed by the desire to provide a blueprint to create programs that could be accredited. We discuss this point in more detail below.

The model curriculum would be developed within the context of the Computing Curricula project (www.acm.org/education/). The Computing Curricula project is a collaboration between the ACM (Association for Computing Machinery), AIS (Association for Information Systems) and IEEE Computer Society and aims to produce model curricula for all computing disciplines, as well as an overview volume that describes the differences between the different curricula (Joint Task Force for Computing Curricula, 2005). The committee decided to follow the format of other documents developed within this project, in particular CC2001 – Computer Science (Joint ACM/IEEE-CS Task Force, 2001), and adopted the terminology in that document to describe the IT body of knowledge.

Despite the rapidly evolving nature of Information Technology, the curriculum should have some longevity. While there is no doubt that the rapidly changing nature of the field poses additional challenges to IT educators (Cohen, 2002), the committee was of the view that a model curriculum that required significant modification every two years or so would be of very limited utility to institutions wishing to use the model curriculum to set up an IT program.

The curriculum should be flexible and the required body of knowledge must be as small as possible. As we said in the introduction, although there is a "family resemblance" between actual IT programs, there are considerable differences between IT programs, often reflecting the different needs to the constituents that the program is seeking to satisfy. The committee was therefore of the view that a model curriculum that allowed different programs to co-exist, while at the same time giving guidance to institutions wishing to establish an IT program was preferable to one that might lead to a "cookie-cutter" approach to designing IT programs.

The curriculum must reflect the relationship of Information Technology to other computing disciplines. IT programs did not evolve in a vacuum but often evolved out of existing programs in computing. They therefore rely on material that is covered in other computing disciplines as well, and the committee was of the view that it would be unwise to design a model curriculum that did not explicitly acknowledge that IT programs shared many aspects with other computing programs.

The curriculum must reflect those aspects that set Information Technology apart from other computing disciplines. At the same time, it was also clear that IT programs have some unique features that set it apart from existing programs in computing. The committee set out to make that the model curriculum also reflected these unique features.

Inputs into the IT Model Curriculum

Apart from the Delphi exercise conducted at CITC-1 and discussed in more detail below, the curriculum committee used two other sources as the main inputs to the model curriculum. The first source was the accreditation criteria for programs in computing and information technology promulgated by CAC of ABET (ABET CAC, 2005). The committee justified its decision based on principle 1 listed above. The second additional source was the work being done within the Computing Curricula project, and in particular the work done on the Computer Science model curriculum. The decision was justified because of the second principle listed above.

The remainder of this subsection discusses each source in some more detail.

The CITC-1 Delphi study

A first important input into the curriculum design process was a Delphi study conducted with participants at CITC-1 to identify curricular topics in an IT curriculum (Lunt et al., 2004). Each participant was given a pad of self-adhesive sticky notes and a felt-tip marker; they were given twenty minutes to generate as many topics as possible. They were instructed to write one topic per sticky note. At the end of that session over 700 notes had been generated. Participants then organized the topics into groups with the freedom to relocate any of the topics. After approximately 30 minutes there was general agreement on the grouping of topics.

After the results of the Delphi exercise were compiled, 34 groupings were identified. Upon further review some of the topics were combined as they were closely related subsets of other topics, resulting in a set of 27 topics areas, which were then further classified into general education topics, core topics, professional topics and supporting topics, as shown in Table 1. The order in which the topics are listed in each column reflects the frequency with which they were mentioned by participants in the Delphi study. Thus, networking (or some related term) was mentioned most frequently among the core topics.

The results were disseminated to the participants for review and additional input was incorporated. While subsequent discussion refined and slightly altered the initial grouping of topics, the primary groupings in the subsequent IT model curriculum bear at least a family resemblance to the original groupings.

Accreditation criteria

The principle that the IT model curriculum should be a blueprint to create programs that could be accredited led to a second important source of inputs to the IT model curriculum formulation

TABLE 1: Topics identified in a Delphi study by participants at CITC-1 as important for an IT curriculum. The order in which the topics appear within each column reflects the frequency with which they were mentioned during the Delphi exercise.

CORE TOPICS	RELATED TOPICS	PROFESSIONAL TOPICS	GEN.EDUCATION TOPICS
Networking	Hardware: architecture & circuits	Human communications	Math
Software	Thinking/prob. solv.	Project management	Holistic
Web systems design	Embedded systems	Teamwork	General education
Database	Information content	Enterprise topics	Physics
Digital communication	System administration	Ethics	
Data security/privacy	Evaluation	Social factors	
Systems design	Graphics	Co-ops	
Human-computer interaction			
User advocacy			

process, namely the Computing Accreditation Commission (CAC) of ABET accreditation criteria for programs in IT became an important input to the curriculum design process (ABET CAC, 2005). IT was unique among the computing disciplines accredited by the CAC of ABET in that it had formulated accreditation criteria before it formulated a model curriculum. In both Computer Science and Information Systems, the formulation of a model curriculum preceded the formulation of accreditation criteria.

The most recent CAC of ABET accreditation criteria (CAC ABET, 2005), formulated with considerable input from SIGITE, distinguishes between General Criteria and Program Criteria. The intention is that any program in computing must meet the general criteria to be accredited, while specific computing programs, such as programs in computer science, must also meet the relevant program criteria, assuming there are any. Currently, there are program specific criteria for computer science, information systems and information technology.

CAC of ABET's general criteria require programs seeking accreditation to formulate program educational objectives, defined as "statements that describe the career and professional accomplishments that the program is preparing graduates to achieve", and program outcomes, defined as "statements that describe what students are expected to know and be able to do by the time of graduation." In addition, a program is required to have a documented assessment process in place to determine the extent to which its graduates meet the program educational objectives and program outcomes. The results of this assessment process must then be used to improve the program. Many of the other accreditation criteria, such as those regarding faculty size, faculty qualifications, and institutional support, then ask institutions to show that the resources available to the program are sufficient to allow students to achieve the program educational objectives and program outcomes.

In many ways, program outcomes then are central when it comes to accreditation. The program outcomes in the ABET CAC accreditation criteria therefore became an important input in the design of the IT model curriculum. While the actual attributes of graduates that are included in the current version of the ABET CAC accreditation criteria are different in format from the ones that were used by the IT model curriculum committee, there has been no significant change in their content.

The current ABET CAC general criteria list a number of attributes that graduates of any computing program are expected to have at the time of graduation, namely:

- An ability to apply knowledge of computing and mathematics appropriate to the discipline
- An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
- An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
- An ability to function effectively in teams to accomplish a common goal
- An understanding of professional, ethical, and social responsibilities
- An ability to communicate effectively
- An ability to analyze the impact of computing on individuals, organizations, and society, including ethical, legal, security, and global policy issues
- Recognition of the need for and an ability to engage in continuing professional development

- An ability to use current techniques, skills, and tools necessary for computing practice.

As said before, the general criteria are supplemented by program criteria. There is a significant difference between computer science and information systems on the one hand, and information technology on the other, in how this supplemental information is provided. Both computer science and information systems add only 1 or 2 attributes of graduates but include additional information on faculty (both require at least some faculty to have a terminal degree in the subject) and on curriculum. Thus, both specify a number of topics that must be covered for a program to be accredited together with some indication about the time required to cover the topics.

In contrast, the Information Technology program criteria only specify a number of additional attributes that graduates from an IT program must achieve. Since the general criteria state that the curriculum in an accredited program must be designed in such a way that a graduate can achieve the program outcomes, the additional program outcomes in the IT criteria thus give additional guidance about additional curriculum requirements in an indirect manner. The additional attributes in the IT criteria are:

- An ability to use and apply current technical concepts and practices in the core information technologies
- An ability to identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems
- An ability to effectively integrate IT-based solutions into the user environment
- An understanding of best practices and standards and their application
- An ability to assist in the creation of an effective project plan.

By only specifying additional attributes of graduates, rather than direct additional curricular requirements, IT has followed the general trend in accreditation. The advantages of this approach are numerous and include an encouragement of program innovation and experimentation. Moreover, attributes of graduates are preferable to curricular guidelines when it comes to communicating with potential employers about the skills and knowledge that they can expect graduates from particular programs to have, and are a much better vehicle for potential employers to express what they are looking for in graduates from a program.

The computing curricula project

A third significant input into the curriculum process was the Computing Curricula Project, and in particular the CC2001 Computer Science Curriculum document (Joint ACM/IEEE-CS Task Force, 2001). First, the curriculum committee decided that the curriculum should not be organized around courses, in the way in which the model curriculum for Information Systems is primarily organized (Gorgone et al., 2003), but around smaller units. The committee was aware of the wide variety of IT programs and was of the view that this diversity was welcome, especially in a fairly new discipline, such as Information Technology. It felt that organizing the model curriculum around a set of courses might stifle innovation, and it therefore decided to organize the curriculum around what one might call "course building blocks", small units that institutions could combine into courses in a way that best suited their specific needs. Second, the committee adopted a slightly modified version of the terminology developed for the Computer Science model curriculum to refer to these course building blocks. In particular, it adopted the term "Knowledge Unit" to refer to a course building block, and the term "Knowledge Area" to refer to a set of related course building blocks. It defined the terms "Knowledge Area" as a particular disciplinary subfield, and the term "Knowledge Unit" as a thematic module within a knowledge area.

The curriculum committee also followed the lead of the Computer Science model curriculum in defining knowledge units in terms of topics and learning outcomes, but decided not to adopt the distinction between core and elective knowledge units. The committee was of the view that there is a distinction between a curriculum core, i.e., knowledge and skills that *every* graduate from an IT program must obtain, and a set of curricular electives, i.e. knowledge that only individuals specializing in a particular sub-area must achieve. There are numerous careers that graduates from IT programs enter. Those careers show an enormous diversity and the knowledge base and skill sets required for each consequently vary widely. Nevertheless, the committee believed that there were significant overlaps in the skills required for all careers in IT. Moreover, while the committee had adopted the principle early on in the curriculum formulation process that the core of the curriculum should be as small as possible, it also wanted to make a model curriculum that could also give institutions guidance about how to set up an IT program, and it was on this basis that the committee decided to follow the lead of computer science and include both core and elective knowledge and skills.

The committee was of the view that making a distinction between core and elective knowledge units was not the appropriate way for IT to make the distinction between core and specialized knowledge. While this approach was tried early on, it quickly became clear that there were many topics that every graduate of an IT program should have knowledge about, and that the difference between core and elective knowledge and skills did not lie in the topics but in the level of knowledge required. For example, the committee was of the view that every graduate of an IT program should be able to install at least one current operating system but that only those specializing in system administration and maintenance should be able to compare various operating systems and recommend a particular operating system to satisfy given needs. In addition to topics, many knowledge units therefore contain both core learning outcomes, which describe skills for which there is a broad consensus that anyone obtaining a baccalaureate degree in the field must acquire, and elective learning outcomes, typically skills that must be acquired by students specializing in the knowledge area with which that unit is associated. While the core and elective outcomes associated with a knowledge unit typically cover the same topics, the depth to which the topic is covered and the skill levels that students are expected to achieve differ significantly between core and elective learning outcomes.

The Computer Science model curriculum also provided important input in a different way. As the committee worked with the topics generated in the CITC-1 Delphi exercise described above, it became clear that there was considerable overlap with the existing Computer Science model curriculum. More relationships between the topic areas were also generated. As an aid in dealing with the complexity of the hundreds of units in tens of proposed knowledge areas, a concept map of the entire space was generated, which included all of the related knowledge areas and units from the model curricula in Computer Science and to a lesser degree Information Systems and Software Engineering.

The knowledge areas were color coded by the source of the language formulation for the KA. Then the concept maps were merged based upon similarity of knowledge area units. This led to consolidation of knowledge areas. The original color coding was maintained, thus giving a trace of the source of the language used to describe the concept. This process also clarified the differences between the other disciplines and the envisioned IT curriculum. For example, no knowledge area in computer science represented what came to be called “integrative programming”. The orientation of the computer science curriculum is the creation of new software components, not gluing together existing applications; it was therefore not a surprise that scripting languages and the creation of interface matching code were not well represented in the computer science model curriculum.

Formulating the IT Model Curriculum

As we said in the introduction to this section, once the curriculum committee had grown to about 25 individuals, a writing subcommittee was appointed with the responsibility of actually producing the model curriculum. While the writing committee had the ultimate responsibility for generating the curriculum document, it did not do so without seeking input from the other members of the curriculum committee, as well as the wider community. Using the various inputs described above, the writing committee determined the list of knowledge areas to be included in the model curriculum, and the knowledge units to be included in each. It also made fairly high-level decisions on the cognitive level to which particular topics should be covered for every IT student, and, where appropriate, for students specializing in this area. The writing committee used a version of Bloom's taxonomy (Bloom, 1956). However, rather than use a normal 6 level hierarchy, the writing committee condensed this to 3, with level 1 covering Knowledge and Comprehension, level 2 Application and Analysis, and level 3 Synthesis and Evaluation. Level 1 cognitive skills therefore covered roughly the recall and understanding of information, level 2 the ability to use and organize information, and level 3 the ability to generate and evaluate new information. The primary reason the committee condensed Bloom's six levels to three was to allow for faster decision making. During early discussions it had become clear that, while it was in general relatively straightforward to reach a decision about whether a certain topic should be covered in sufficient depth say allow students to be able to recall and understand the concept, it was much harder to reach agreement about whether the student should merely be able to recall the concept, or should be able to demonstrate understanding as well. Moreover, since these preliminary decisions were not intended to be final (see below), the committee believed that there was a premium on decision speed.

In deciding which level of cognitive skill to associate with a particular topic, the writing committee was guided both by its own perceptions of the importance of particular topics, but also by the number of lecture hours that it has associated with each knowledge area. In this regard, it had again taken its lead from the Computer Science model curriculum. First, it had decided that it wanted its core curriculum to be roughly the same size as the Computer Science model curriculum. The Computer Science core curriculum consists of 280 core hours, and the writing committee therefore assigned a number of hours to each knowledge area so that the total number of hours added to 281. Second, the writing committee also followed the lead of the Computer Science model curriculum in using the lecture hour as its primary unit of measurement. The committee made an effort to point out that it did not endorse the lecture method as the only, or for some topics even the preferred, method of covering material. It merely used it as a unit of measurement.

Presented with the list of knowledge areas and knowledge units, the curriculum committee established a separate subcommittee for each knowledge area with the task of formulating a set of core and elective learning outcomes for each unit, using the initial determination of the required cognitive level both for all students and specialists as its guide. The subcommittees were given the authority to re-organize the knowledge area but they were strongly urged not to increase the number of core hours for each knowledge area.

Table 2 is an example of part of the information that the subcommittee responsible for formulating the Information Management knowledge area was given. Line 2 indicates that the writing committee was of the view that every student in an IT program should develop level 1 cognitive skills for the topic of ER diagrams, while students specializing in data management should achieve level 3 cognitive skills. Table 3 shows the resulting knowledge unit in the IT model curriculum. The knowledge unit is entitled "Data Modeling" and the writing committee had initially decided to associate 3 core hours with this unit. As Table 3 shows, the subcommittee responsible for the Information Management knowledge area was of the view that this particular knowledge unit needed more core hours. Table 3 also illustrates the difference between the different cognitive levels. For example, the level of cognitive skills concerning entity-relationship diagram is fairly low for all IT students but considerably higher for information management specialists. Similarly, while there is no mention of the reengineering of databases among the core learning outcomes, reflecting the cognitive level of 0 assigned to this subtopic by the writing committee, there is an elective learning outcome associated with this topic.

TABLE 2: Part of the information given by the writing committee to the Information Management subcommittee for the knowledge unit Data Modeling.

TOPIC	SUBTOPIC	COGNITIVE LEVEL	
		Core	Spec
Conceptual models			
	Entity relationship diagrams	1	3
	Enhanced ER diagrams	1	3
	Identification of business rules	2	3
Logical models		1	3
Physical models		1	3
Reengineering of databases		0	2
Standardized modeling In UML		1	3
Meta-modeling		1	3
Data integration		1	2

TABLE 3: The Data Modeling Knowledge Unit in the IT Model Curriculum.

IM4: Data Modeling
Minimum core coverage time: 6 hours
Topics:

- Conceptual Models
  - Entity Relationship diagrams
  - Enhanced Entity Relationship diagrams
  - Identification of business rules
- Logical Models
- Physical Models
- Reengineering of Databases
- Standardized Modeling in IDEF1, UML
  - Patterns and standard models
- CASE Tools
- Meta-modeling
- Data integration
  - Data warehouses
  - Data marts

Core learning outcomes:

1. Describe and interpret Entity Relationship diagrams;
2. Design a simple Entity Relationship diagram;
3. Describe and interpret Enhanced Entity Relationship diagrams;
4. Identify business rules;
5. Describe a logical model;
6. Describe a physical model;
7. Identify patterns and standard models;
8. Demonstrate an understanding of CASE tools, their usage and application;
9. Describe data integration;
10. Describe meta-modeling;
11. Describe a data warehouse, its basic structure, etc.;

Elective learning outcomes:

1. Create and design Entity Relationship diagrams;
2. Create and design Enhanced Entity Relationship diagrams;
3. Formulate and explain identification of business rules;
4. Create and evaluate a logical model;
5. Create and evaluate a physical model;
6. Demonstrate how to reengineer databases;
7. Create and compare patterns and standard models;
8. Use a given CASE tools;
9. Create and evaluate meta-models;
10. Explain the concept of data integration and its use in the creation of data warehouses and data marts;
11. Change an existing data warehouse;
12. Change an existing data mart.

Once the various subcommittees had completed their tasks, the writing committee, after a fairly careful inspection, included the expanded knowledge areas in the IT model curriculum document. The first draft of the IT model curriculum document was produced in the summer of 2004. It was presented at SIGITE 2004, at which time the committee sought feedback from the IT community present at this conference. The document was also made available for public comment via the ACM web site. A revised version of the IT model curriculum was posted in April 2005, again with a message soliciting additional feedback. The feedback was incorporated in a slightly revised version that was published in October 2005 (SIGITE Curriculum Committee, 2005).

Results

Knowledge Areas in the IT Curriculum

The IT model curriculum contains the following knowledge areas:

Information Technology Fundamentals (ITF)

Human Computer Interaction (HCI)

Information Assurance and Security (IAS)

Information Management (IM)

Integrative Programming & Technologies (IPT)

Networking (NET)

Programming Fundamentals (PF)

Platform Technologies (PT)

Systems Administration and Maintenance (SA)

System Integration & Architecture (SIA)

Social and Professional Issues (SP)

Web Systems and Technologies (WS)

The rationale for the inclusion of many of these modules follows directly from the definition of Information Technology as focusing on meeting the needs of users within an organizational and societal context through the **selection, creation, application, integration and administration** of computing technologies. In order to be able to do so, an IT graduate must develop a myriad of skills.

First, every IT graduate must develop knowledge of the hardware and networking. This is covered in the knowledge areas Networking (NET) and Platform Technologies (PT). NET covers data communications, telecommunications, inter/intranetworking, infrastructure and an introduction to security. It also includes application of networking to multimedia, information storage and distribution, and the World Wide Web. PT covers the fundamentals of hardware and software and how they integrate to form essential components of IT systems.

Second, as with any computing professional, the IT graduate must develop the skill to program, and there are two knowledge areas that cover this, namely Programming Fundamentals (PF) and Integrative Programming and Technologies (IPT). PF is very similar to programming courses or knowledge areas in the other computing model curricula. It develops skills and concepts that are essential to good programming practice and problem solving, and covers fundamental programming concepts, event-driven programming, object-oriented programming, basic data structures, and algorithmic processes. In contrast, IPT is unique to the IT model curriculum and reflects the fact that the IT professional is more likely to be asked to integrate existing components, rather than develop new components. This knowledge area helps students develop the skills to do so and examines the various types of programming languages and their appropriate use, and the use of scripting languages, architectures, application programming interfaces and programming practices to facilitate the management, integration and security of the systems that support an organization.

Third, an important task of many IT professionals is to design, select, apply, and deploy computing systems and integrate them into the organization. They must also be able to administer and maintain these systems. This set of skills is developed in the System Integration and Architecture (SIA) and System Administration and Maintenance (SA) knowledge areas. SIA develops the skills to gather requirements, source, evaluate and integrate components into a single system, and validate the system. It also covers the fundamentals of project management and the interplay between IT applications and organizational processes. SA covers the skills and concepts that are essential to the administration of operating systems, networks, software, file systems, file servers, web systems, and database systems, as well as skills related to system documentation, policies, and procedures. It also includes education and support of the users of these systems.

Fourth, IT professionals focus on meeting the needs of users. It is therefore important that they develop the skill to determine user needs. Although aspects of this are covered in SIA, and indeed throughout the curriculum, the user is the primary focus of Human Computer Interaction (HCI). HCI is designed to help graduates develop a mind-set that recognizes the importance of users and organizational contexts and to learn to employ user-centered methodologies in the development, evaluation, and deployment of IT applications and systems. It covers such areas as user and task analysis, human factors, ergonomics, accessibility standards, and cognitive psychology. However, in order to adequately meet the needs of users, IT professionals must also develop an awareness of the social and professional context of information technology and computing, and adhere to ethical codes of conduct. The Social and Professional Issues (SP) knowledge area allows them to do so. It covers the historical, social, professional, ethical, and legal aspects of computing. It also identifies how teamwork is integrated throughout IT and how IT supports an organization. Finally, it stresses professional oral and written communication skills.

Fifth, IT graduates must develop technical skills to develop and maintain modern IT applications. Since most of these use some underlying database and/or are web-based, IT graduates must develop skills related to the management of information and web technologies. These skills are covered in the Information Management (IM) and Web Systems and Technologies (WS) knowledge areas respectively.

The final set of skills that IT professionals have to develop relate to security. IT systems are increasingly under attack. The IT professional must understand, apply, and manage information assurance and security in computing, communication, and organizational systems. The Information Assurance and Security (IAS) knowledge area is designed to allow them to develop these skills. It also instructs students how to provide users with a framework to be sufficiently security aware in order to be an asset to the organization rather than a liability. IAS includes operational issues, policies and procedures, attacks and defense mechanisms, risk analyses, recovery, and information security.

Pervasive Themes

As the committee was working on the different knowledge areas, which are to some extent independent, the committee noted that there were a number of themes that pervaded the curriculum. The notion of a pervasive theme is similar to that of a "recurrent concept", which Alford, Carter, Ragsdale, Ressler, & Reynolds (2004) point out, first appeared in the computer science model curriculum in 1991. The notion is related to what Wiggins and McTighe (1998) call "enduring understanding", a set of "big ideas" that reside at the heart of the discipline and that cannot be covered directly but must somehow be grasped by students as they become proficient in the discipline.

Although IT will share some of these pervasive themes with the other computing disciplines, their combination is unique to IT and sets IT apart from the other computing disciplines. The pervasive themes are:

User centeredness and advocacy

Information assurance and security

The ability to manage complexity through: abstraction and modeling, best practices, patterns, standards, and the use of appropriate tools

A deep understanding of information and communication technologies and their associated tools

Adaptability

Professionalism (life-long learning, professional development, ethics, responsibility)

Interpersonal skills

These pervasive themes are expected to recur throughout the curriculum. However, the model curriculum also proposes that they are explicitly introduced in a first IT course that majors are exposed to. This course is intended to cover the IT Fundamentals (ITF) knowledge area. This knowledge area provides an overview of the discipline of IT, describes how it relates to other computing disciplines, and begins to instill an IT mindset. The goal is to help students understand the diverse contexts in which IT is used and the challenges inherent in the diffusion of innovative technology.

The Model Curriculum as a Guide

The model curriculum uses a number of methods to give guidance to institutions wishing to implement the model curriculum. First, each knowledge area specifies the number of hours that the writing committee expects to be necessary to cover the core learning outcomes. As said before, the committee followed the lead of computer science and used lecture hours as the unit of measurement but explicitly stated that it did not endorse lectures as the only or the best way to cover material. Second, the IT model curriculum associates core and elective learning outcomes with many of the knowledge units. As said before, core learning outcomes are expected to be achieved by all students graduating from an IT program, whereas elective learning outcomes are examples of outcomes expected to be achieved only by those specializing in that particular knowledge area. Table 4 provides more details about the structure of the IT model curriculum:

As one would expect, certain knowledge areas contain no elective learning outcomes. For example, given the foundational nature of the IT Fundamentals (ITF) and Programming Fundamentals (PF) knowledge areas, it should come as no surprise that there are no elective learning outcomes associated with these knowledge areas. The belief is that all IT students, no matter what their specialization, must achieve the skills described in these knowledge areas. Similarly, since it is hard to see how an IT professional could specialize in Social and Professional Issues (SP), there are no elective learning outcomes associated with this knowledge area.

On the other hand, the fact that the other knowledge areas contain both core and learning outcomes reflects the belief of the IT curriculum committee that it is possible for graduates from IT programs to specialize in each of these areas. Thus, it is entirely feasible for an IT student to specialize in Networking (NET) and/or Information Management (IM). However, the committee was also of the view that every graduate from an IT program should develop some skills in each of these areas. Also, the committee did not want to imply that IT students should specialize in only one knowledge area. In fact, it seems more logical for students to specialize in clusters of knowledge areas. One natural cluster might be Networking, Information Assurance and Security, and System Administration. Another natural cluster might be Information Management, Integrative Programming Techniques and Web Systems and Technologies, or a combination of Human Computer Interaction and System Integration and Architecture. However, given that the commit-

TABLE 4: Structure of the IT Model Curriculum

KA	# Core Hours	# KUs	# Core Outcomes	# Elective Outcomes	# KUs with core	# KUs with electives	# KUs with core only	KUs with electives only
ITF	33	6	35	0	6	0	6	0
HCI	20	7	35	26	7	7	0	0
IAS	23	11	89	49	11	11	0	0
IM	34	6	52	44	6	6	0	0
IPT	23	7	26	17	7	6	1	0
NET	20	6	44	30	5	5	1	1
PF	38	6	29	0	6	0	6	0
PT	14	6	13	16	3	6	0	3
SA	11	4	28	11	4	2	2	0
SIA	21	7	44	29	7	7	0	0
SP	23	9	49	0	9	0	9	0
WS	21	6	52	51	5	6	0	1

tee wanted to allow programs a great deal of flexibility, it stopped short of making definite recommendations about how to cluster knowledge areas.

Finally, although the core of the IT model curriculum consists of knowledge areas, the curriculum document also contains a number of examples of courses that an institution might offer. However, the only firm recommendation that the curriculum document contains regarding the implementation of the curriculum is that most of the material contained in IT Fundamentals (ITF) and Programming Fundamentals (PF) should be covered early in any academic program of study. The reason for the recommendation that PF be covered early in the curriculum is that the material in this knowledge area is prerequisite for a number of other knowledge areas, including Integrative Programming Techniques, Information Management and Web Systems and Technologies. The reason for the recommendation that ITF be covered early in the curriculum is that the material contained in this knowledge area sets the stage for any IT program in that it defines how IT differs from other computing disciplines, and introduces the themes that are recurring throughout the remainder of the curriculum.

Conclusion

This paper has presented the current IT model curriculum, and described the process by which the document was formulated. It also justified the various decisions that were made along the way, both to streamline the process that was followed to create the document, and in the design of the document itself. One of the primary motivations was to produce a model curriculum that was consistent with the emerging accreditation standards for programs in IT, and that built on existing computing curricula while at the same time clarifying the differences between IT and the other computing disciplines. Moreover, given the fact that there are many different "flavors" of IT programs, the committee had a strong desire to produce a model curriculum that on the one hand provided sufficient guidance to institutions interested in establishing an IT program, while on the other hand allowing institutions sufficient flexibility to design IT programs that best reflect their institutional culture and the specific needs of their stakeholders. The committee is of the view that the current document meets the various requirements that it formulated from the outset. The document provides a blue print for institutions wishing to design programs that are accreditable without forcing them into a curricular straightjacket.

However, the committee also recognizes the dynamic nature of computing and specifically the IT environment and recommends that there be an ongoing review process that allows individual components of the curriculum recommendations to be updated on a recurring basis. The full IT model curriculum is available at http://www.acm.org/education/curric_vols/IT_October_2005.pdf. The site also allows interested readers to comment on the current version. We want to end by expressing the hope that many readers will leave constructive comments as we strive to further develop a fundamentally strong, diversified, outcome driven curriculum for IT.

References

ABET CAC. (2005). Criteria for accrediting computing programs. Baltimore, MD. Available at www.abet.org/forms.shtml

Alford, K., Carter, C., Ragsdale, D., Ressler, E., & Reynolds, C. (2004). Specification and managed development of information technology curricula. *Proceedings 5th Conference on IT Education*, Salt Lake City, UT, 261-266.

Bloom, B. (Ed.). (1956). *Taxonomy of educational objectives: Book 1, Cognitive domain*. New York: Longman.

Cohen, E. (Ed.). (2002). *Challenges of information technology in the 21st century*. Hershey, PA; Idea Group.

Dalhom, D., & Mathiassen L. (1997). The future of our profession. *Communications of the ACM, 40*(6), 80.

Denning, P.J. (2001a). The IT schools movement. *Communications of the ACM, 44*(11), 18.

Denning, P.J. (2001b). Who are we? *Communications of the ACM, 44*(2), 15.

Denning, P.J. (2004). The field of programmers myth. *Communications of the ACM, 47*(7), 15.

Gorgone, J., Davis, G., Valacich, J., Topi, H., Feinstein, D., & Longenecker, H. (2003). IS 2002 model curriculum and guidelines for undergraduate degree programs in information systems. *Communications of the AIS, 11*, Article 1.

Joint ACM/IEEE-CS Task Force. (2001). Joint ACM/IEEE-CS task force on computing curricula 2001 for computer science, Final report. Retrieved from http://www.computer.org/education/cc2001/final/index.htm

Joint Task Force for Computing Curricula. (2005). Overview report. Retrieved from http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf

Lunt, B., Reichgelt, H., Ashford, T., Phelps, A., Slanzski, E., & Willis, C. (2003a). An empirical comparison of baccalaureate programs in computing. *Proceedings of the 2003 International Conference on Engineering and Computer Education (ICECE)* Mar 17-20, 2003, Santos, SP, Brazil. [Proceedings on CD-ROM (no page numbers).]

Lunt, B., Reichgelt, H., Ashford, T., Phelps, A., Slanzski, E., & Willis, C. (2003b). What is the new discipline of information technology? Where does it fit? *Conference for Industry and Education Collaboration (CIEC) 2003*, Jan 28-31, 2003, Tucson, AZ. [Proceedings on CD-ROM (no page numbers).]

Lunt, B., Ekstrom, J. Gorka, S., Kamali, R., Lawson, E., Miller, J., & Reichgelt, H. (2004). Defining the IT curriculum: The results of the past 2½ years. *Proceedings of the International Conference on Engineering Education and Research (iCEER),* June 27-30, 2004, Olomouc, Czech Republic. 2004; Proceedings on CD-ROM (no page numbers)

Lunt, B., Ekstrom, J. Gorka, S., Kamali, R., Lawson, E., Miller, J., & Reichgelt, H. (2005). Defining the IT curriculum: The results of the past 3 years. *Journal of Issues in Informing Science and Information Technology, 2*, 259-270. Available at http://2005papers.iisit.org/I21f61Lunt.pdf

Reichgelt, H., Lunt, B., Ashford, T., Phelps, A., Slanzski, E., Willis, C. (2004). A comparison of baccalaureate programs in information technology with baccalaureate programs in computer science and information systems. *Journal of Information Technology Education, 3*, 19-34. Available at http://jite.org/documents/Vol3/v3p019-034-098.pdf

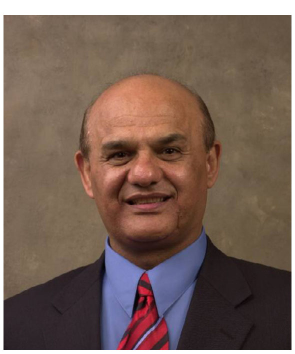
SIGITE Curriculum Committee. (2005). IT 2005 model curriculum report, October 2005. Retrieved from http://www.acm.org/education/curric_vols/IT_October_2005.pdf

Wiggins, G. & McTighe, J. (1998). *Understanding by design*. Alexandria, VA: Association for Supervision and Curriculum Development.

Biographies

Joseph J. Ekstrom (Ph. D. Computer Science, BYU 1992) has been Associate Professor of Information Technology at BYU since 2001. During 30 years of industrial experience he held positions from developer through senior management. His research interests include network and systems management, distributed computing, system modeling and architecture, system development, Information Retrieval, and IT curriculum and instruction.

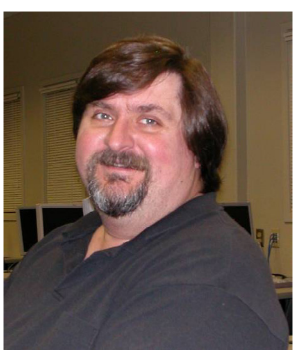
Sandra Gorka is an Assistant Professor of Information Technology at the Pennsylvania College of Technology (PCT) in Williamsport, Pennsylvania. Dr. Gorka received a B.A. in mathematics from Shippensburg University, and an M.S. degree and Ph.D. in mathematics and topology from the University of Delaware. She has worked for several years as a private consultant. Dr. Gorka has been involved with the ACM Special Interest Group for Information Technology Education (SIGITE) and was a founding member of its predecessor Society for Information Technology Education (SITE). Her present research interests include IT curriculum, pedagogy and education.



Reza Kamali, PhD, is an Associate Professor and department head of Computer Information Technology at Purdue University Calumet, Hammond, IN. He was a founding member and Education Officer of SITE, which later became ACM's SIGITE. Dr. Kamali was a member of The Joint Task Force for Computing Curricula 2005 and a member of IT 2006 Task Force.

Ms. **Eydie Lawson** has an MS in Computer Science and an MS in Business from the Rochester Institute of Technology (RIT). She taught business and economics for over 20 years and came over to the world of computers in 1986 to start a new career. She joined the Computer Science Department at RIT in 1990 and was the Assistant Director of the School of Computer Science and Information Technology from 1990-1995. She moved to the Information Technology Department in 1993 and was the Department Chair 1995-2003. While she was Chair, she guided the department through its explosive growth. She currently serves as Associate Dean of the College of Computing and Information Sciences. Ms. Lawson is one of the founding members and past chair of the ACM Special Interest Group (SIG) for Information Technology (SIGITE) and is currently a member of the ACM Education Council and a member-at-large of the ACM SIG Governing Board Executive Committee. Working with SIGITE and ABET, Eydie has been pivotal in the development of the accreditation criteria and model curriculum for Information Technology. She has numerous presentations and publications and was named IT Woman of the Year for 2003 for Upstate New York by the Association for Women in Computing.

Barry M. Lunt is an Associate Professor of Information Technology at Brigham Young University in Provo , UT. Dr. Lunt received a B.S. and an M.S. degree in EET from BYU, and a Ph.D. in Occupational and Adult Education from Utah State University in Logan , UT. He has spent seven years in industry as a design engineer. His present research emphases are the physical design of electronic circuits and systems, IT curriculum, and engineering education.



Jacob Miller is an Associate Professor of Information Technology at the Pennsylvania College of Technology in Williamsport, Pennsylvania. His qualifications include an M.S. and Ph.D. in Mathematics (dynamical systems) from the University of Delaware. Additionally, he spent seven years working for IBM in inventory planning and logistics and spent several years as a consultant for various organizations. Dr. Miller has been involved with the ACM Special Interest Group for Information Technology Education (SIGITE) and was a founding member of its predecessor Society for Information Technology Education (SITE). Dr. Miller's research interests include IT education and curriculum development, networking, information assurance and security, and developing experiential learning activities for IT students.

Han Reichgelt holds degrees in Philosophy and Psychology from the University of Nijmegen in the Netherlands, and a PhD in Cognitive Science from the University of Edinburgh in Scotland. Previously, he was a research fellow at the University of Edinburgh, lecturer in Psychology at the University of Nottingham, and professor of Computer Science at the University of the West Indies, Mona, Jamaica. His current position is Associate Dean of the College of Information Technology at Georgia Southern University. Han is the (co-) author of over 100 journal and refereed conference papers, as well as a textbook on Knowledge Representation in Artificial Intelligence. More recently, he co-edited a book on Information Systems Delivery Quality. His research interests include IT and economic development, IT application delivery quality, business process modeling and computing education. He currently chairs the ACM Special Interest Group on IT Education.