December 2003

How Much Is Enough? Teaching Information Technology in a Business-Oriented IS Curriculum

John Beachboard
Idaho State University

Kevin Parker
Idaho State University

How Much Is Enough? Teaching Information Technology in a Business-Oriented IS Curriculum

John C. Beachboard
Idaho State University
beach@isu.edu

Kevin R. Parker
Idaho State University
parkerkr@isu.edu

Abstract

IS 2002 is a model curriculum for undergraduate degree programs in Information Systems typically offered in business and management schools. The curriculum is designed to produce graduates equipped to assume entry-level information systems positions and provide them with the fundamental knowledge required for continued career growth. This paper specifically addresses the challenge that instructors face in designing a course intended to be consistent with the learning unit goals and objectives specified in the IS 2002 for the Information Technology Hardware and Software course. While the authors agree in principle with the identified course objectives, they suggest that those objectives, when broadly defined, constitute more material than can be reasonably covered in a single course. Given that some depth must be sacrificed to achieve the desired breadth of coverage, the paper calls for the IS community to participate in an effort to more fully specify concepts and skills that should be taught in order to best fulfill our students' and their future employers' needs. The paper describes and critiques the authors' current course design as a starting point for this effort.

Introduction

How technical do graduates of undergraduate IS programs need to be to successfully compete for jobs in today's competitive markets? Do business-oriented IS programs[1] provide adequate levels of technical education? The *IS 2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems*, a joint effort of the ACM, AIS, and AITP societies, represents the most recent product of an effort that began in the early 1970s to assist universities in developing degree programs that will produce "graduates equipped to function in entry-level information systems positions with a strong basis for continued career growth (2002, p. vi). This most recent iteration of the curriculum report indicates that the typical business school student graduating with an MIS or CIS emphasis may lack the technical skills today's IS employers are seeking. Therefore, the proposed curriculum is intended to respond to industry requests for "both increased emphasis in technical orientation and improved skill in individual and group interactions (2002, p. vi).[2]

While IS 2002 addresses a broad range of curriculum issues, this paper focuses specifically on the issues of defining and achieving a desired level of technical competency for graduates from business-oriented IS programs. Rather than suggesting that we have successfully addressed these issues, the authors hope that this paper will help initiate a dialogue aimed at further defining what

[1]We use the term "business-oriented IS" to refer to information systems programs offered in AACSB-accredited colleges of business as opposed to IS-related offerings from other colleges which are not subject to the same accreditation restrictions.

[2]IS 2002 is an update of IS '97. In the interest of brevity, the authors have omitted much background information included in the report. The full report is available at **http://www.acm.org/education/curricula.html#IS2002**.

constitutes an appropriate breadth and depth of coverage for courses designed to satisfy the learning objectives of IS 2002.4 -- Information Technology Hardware and Systems Software.[3]

The Issues

The IS 2002.4 course specification is intended to provide the "hardware/software technology background to enable systems development personnel to understand tradeoffs in computer architecture for effective use in a business environment. System architecture for networked computing systems and operating systems will be covered" (Gorgone, et al., 2002, p. 26). The IS 2002.4 course is intended to combine lectures presenting the theoretical underpinnings of the various elements of information systems installation and operational laboratory exercises. Table 1 lists the learning unit goals specified for IS 2002.4.

Table 1. IS 2002.4 – Information Technology Hardware and System Software (Gorgone, et al., 2002, p. 44)

Learning Unit Number	Learning Unit Goal
62	to explain in systems terms the fundamental characteristics and components of computer and telecommunications hardware, and system software, and demonstrate how these components interact
63	to provide an overview of peripheral devices and their function
64	to introduce the concepts of computer hardware architectures
65	to introduce the concepts of system software components and interactions
67	to introduce the major concepts in operating systems, including process definition, concurrent processing, memory management, scheduling, interrupt processing, security, and file systems
68	to introduce a variety of operating environments (traditional, GUI, multimedia) and resource requirements
69	to discuss, explain, and install multimedia facilities
70	to introduce the requirements for interoperability and systems integration
71	to install, configure, and operate a multi-user operating system

We see some difficulty in successfully presenting our students with both the theory and practical skills implied by the goals identified in Table 1. These unit goals present two challenges. The first challenge concerns the sheer amount of theory and concepts that we want the students to absorb. Units 62 through 64 and 70 seem largely consistent with the objectives of a typical three-credit-hour course in systems architecture required by many computer science programs. Units 65 through 68 reflect the topics taught in a separate three-credit-hour course on operating systems. Admittedly, there must be an expectation that this course will cover a subset of the content offered in the corresponding six credit hours of computer science courses. The danger is that attempting to present too much material in a single course will cause student comprehension and retention to suffer.

The second challenge concerns the evident desire to provide students with practical applications of referenced technical concepts as implied by the inclusion of the learning unit 71. This learning unit establishes the objective for students to be able to install, configure and operate a multi-user operating system. We recognize that this does not necessarily imply that students are to become certified Windows or Unix systems administrators, but we are concerned that the inclusion of this objective within the IS 2002.4 course as outlined can be viewed as trivializing the level of knowledge required to meaningfully understand and perform systems administration tasks associated with configuration and operation of multi-user operating systems. The need to provide meaningful laboratory exercises requires additional time and would be affected by staffing and facility constraints.

[3]The authors recognize that universities may support employment markets with various needs, but agree with the report's implicit assumption that there are benefits to a reference model curriculum built around "a fundamental body of computing and information systems knowledge" (Gorgone, et al., 2002, p. v).

If the theoretical and practical course goals for IS 2002.4 are broadly defined, they appear to exceed what can be reasonably accomplished in a three-semester-hour course. Yet, if we narrow our interpretation too much, we risk undermining the intent of the revised curriculum to increase the technical orientation of the IS program of study. We would like to see the IS community develop a consensus regarding what constitutes the knowledge and skills most critical for our students to acquire from this course.

In the next section we describe our current course offering that is intended to satisfy the goals established for IS 2002.4. While we recognize limitations in our current course design, the following section provides a fairly detailed course description to serve as a point of departure for a discussion aimed at building such a consensus.

Introducing Software and Systems Architecture

Our university, which is a medium-sized public university in the Northwestern United States, offers a two credit-hour course intended to satisfy most of the learning objectives associated with IS 2002.4. The course is offered to freshman and sophomore students prior to or concurrent with their first programming course. The Computer Information Systems (CIS) Department wanted to provide students with a fundamental understanding of how computer systems work before or during their first programming course. We felt that this would help level the playing field for students entering the CIS program with relatively little experience with computers. Therein also lies a challenge. We are introducing students with widely varying levels of computer expertise to fairly technical concepts. This approach has required us to consider carefully the core technical concepts that we wish our students to understand upon completing this course.

Our course, as currently offered, is a two-credit-hour course that includes 11 learning units. The course is structured around Stephen Burd's (2001) text Systems Architecture but incorporates materials drawn from many additional sources. The following paragraphs describe the primary concepts presented in each learning unit and identify their relationship to the learning goals listed at Table 1.

Introduction to General and IS Systems [Learning Unit 62]. Students are introduced to systems theory as a means of dealing with or conceptualizing complex phenomena as exemplified by enterprise-class information systems employed in complex business organizations. We expose students to the importance of perspective and purpose in defining systems and, in a very rudimentary sense, present the laws of composition and decomposition. Students are then introduced to information systems as being comprised of people, processes, information and technology. We stress that the primary focus of this course is on the technology but that we want students to remain sensitive to the broader context in which the technology is employed. We define and contrast the concepts of IS application and IS infrastructure and discuss the distinction between them. Finally, we introduce the notion of IS architecture in terms of technical choices and argue that knowledge of technology is required to make sound technical choices in designing an organization's IS infrastructure. Much of material for this section is drawn from Weill and Broadbent's (1998) *Leveraging the New Infrastructure: How Market Leaders Capitalize on Information Technology.*

Applications Development [Learning Units 62, 70 and 71]. This learning unit is comprised of two distinct segments. Again, our goal is to reinforce the notion that it is the application of technology, rather than technology itself, that we are interested in. Accordingly, we first discuss various types of business applications and introduce the creation of software objects to represent physical objects and processes in the real world. Typically, this concept is presented in terms of what is accomplished when creating an airline or concert hall seat-reservation system. Secondly, students perform a very simple programming exercise that is used to introduce the concept of assignment statements and instruction sequence in the development of applications. We also discuss the function of interpreters and compilers, and assign a small exercise that demonstrates their impact on program performance. A short programming assignment is included with this learning unit.

Computer Systems Architecture Overview and Introduction to Operating Systems [Learning Units 62, 64, 65 and 67]. In this learning unit we provide a textbook definition of a general-purpose computer system and identify its major components. However, the fundamental focus of this set of lectures is to help students conceptualize computer systems as consisting of a layered architecture. We discuss layers and modularization as a means of dealing with complex problems, and we emphasize how the application of this design technique can ease follow-on system maintenance. We introduce operating systems as a component in this layered architecture and describe the two primary functions or roles of an OS. The first role is to act as an extended (or virtual) machine (Tanenbaum, 2001, pp. 3-4). We compare and contrast operating systems with the Java virtual machine environment to illustrate multiple approaches to representing processing environments in software, explaining how these virtual representations support application portability. The second operating system function is system resource management. In this discussion we introduce process and memory management techniques (Tanenbaum, 2001, pp. 5-6; Burd, 2001, pp. 409-432).

While in no way approaching Tanenbaum's level of detail and rigor, this learning unit draws heavily on the perspective Tanenbaum employs in organizing his computer science texts *Structured Computer Organization* (1999) and *Modern Operating Systems* (2001).

Data Representation [Learning Unit 62]. This learning unit introduces students to techniques used to translate from the symbol systems that are commonly used into a symbol system that can be understood and manipulated by computers. The key elements emphasized are design decisions relating to compactness, accuracy, ease of manipulation and standardization. Students are introduced to primitive data types and relatively simple data structures (Burd, 2001, p. 71). They are expected to become competent in converting binary numbers. We spend considerable time is analyzing the format of the IEEE 32-bit floating-point data type. In describing how bits are allocated between representing the exponent and mantissa, we emphasize how designers make explicit choices in developing data representation to balance competing data representation goals such as range and accuracy. The concepts of arrays and linked list data structures are discussed, as are the tradeoffs between the two with respect to storage efficiency and system performance.

Processor Architecture [Learning Units 62 and 64]. The challenge of this unit is to introduce students to fundamental concepts governing the design and performance characteristics of processor architectures without becoming too immersed in technical details. We begin by introducing (or reintroducing) students to Boolean logic and demonstrating how it provides the basis for digital logic circuitry. Students are expected to understand how Boolean logic is used in the design of circuitry that performs simple integer addition. In our view the implementation details are less important than giving the students a concrete example of how symbolic logic functions are instantiated in physical devices. This helps us to explain the relationship among bits, bytes and words, and helps students to understand how word size and the support of primitive data types are implemented in hardware. An abbreviated explanation of basic electronics is provided to explain internal signaling and the significance of processor clock cycles. The unit closes with a relatively brief discussion of other design features impacting performance such as parallelism, pipelining, branch prediction, out-of-order processing, and inclusion of circuitry to perform specialized (e.g., multimedia) functions.

Storage Architecture [Learning Units 62, 64 and 63]. The unit on storage architecture emphasizes the practical implications of the memory and storage hierarchy with respect to storage speed, volatility, cost and capacity. The technical aspects of SRAM, DRAM, SDRAM and RDRAM are briefly presented so that students will have an understanding of the benefits of cache memory and the various types of systems memory available. The discussion primarily concerns the speed and performance tradeoffs for various types of systems memory as well as emphasizing the tremendous performance differences between primary and secondary memory. The function of memory controllers is discussed in the context of introducing logical and physical memory addressing. Magnetic disk drive technology is introduced and includes a brief description of the low-level formatting process and some determinants of disk drive performance. We briefly identify, rather than truly explain, the differences in disk controller technology and examine the cost, performance and reliability characteristics of RAID 0, RAID 1 and RAID 5. We finish this section by revisiting the discussion of memory hierarchy by contrasting memory caching with the configuration of virtual memory.

Systems Integration and Performance [Learning Units 62, 63 - 65 and 67]. The systems-integration learning unit introduces students to motherboard and bus technologies. We present the differences between parallel and serial bus architectures and briefly discuss bus protocols and centralized versus decentralized bus arbitration techniques. The emphasis is placed on cost and performance considerations, and we expect students to appreciate tradeoffs made when system architects choose to incorporate multiple bus technologies in their system designs. We reiterate the function of device controllers and discuss how they may impact system performance. The unit closes with a discussion of system interrupt processing, the benefits of buffering, and the distinction between buffers and caches.

Networking [Learning Units 62, 64 and 70]. The networking learning unit has been the most difficult to formulate. While CIS majors are required to take a full networking course, this will be the last IS infrastructure-oriented course for many students. The challenge is to give these students a sufficient appreciation for networking technology without covering too much material that will be repeated in the networking course. We provide a conceptual overview of communications systems as consisting of transmission media, switches and protocols. We contrast mesh and star topologies to illustrate how switches are used to reduce the demand for relatively expensive transmission media and network connections, and we emphasize the tradeoffs among cost, performance and reliability inherent in the design of any communications architecture.

We explain the differences between circuit- and packet-switching technology. We describe the physical and logical architecture of the Internet, including a brief description of the performance and reliability (survivability) characteristics that encouraged the military's investment in the development of packet-switching technology. Students are introduced to the Internet protocol suite,

although the technical details provided are minimal. We describe the socket interface to the protocol stack and identify TCP as an application-to-application protocol and the layer in which the data stream is divided into packets. We also explain in a very rudimentary manner the composition of an IP address, DNS and IP routing.

Performance Benchmarking [Learning Units 62 and 70]. We wrap up our introduction to the various types of hardware technologies with a lecture on benchmarking. We explain common computer benchmarking tools and measures, including MIPS, GFLOPS, Kernel programs, toy benchmarks, synthetic benchmarks, and application benchmarks. We then discuss each in terms of their respective strengths and limitations. For example, we explain how manufacturers can optimize systems to perform well on benchmarks while not necessarily improving practical performance. Finally, we demonstrate several benchmarking tools such as SiSoftware's Sandra (The System ANalyser, Diagnostic and Reporting Assistant) and encourage students to download and try them out on their own systems.

Operating and File Management Systems [Learning Units 62, 65, 67, 70 and 71]. Once students have a better understanding of the various hardware components of a computer system, we review the two primary operating system functions described earlier. We re-emphasize the layered design of modern operating systems and suggest a parallel-layered function and design of file management systems. We review low-level formatting and then explain the creation of disk partitions, emphasizing the distinction between disk partitioning and high-level formatting. We present reasons why a systems administrator might want to partition system disks with respect to the selection of cluster or allocation unit size and the function of file allocation tables. We explain directory content and structure, and discuss potential performance implications associated with the selection of an appropriate directory structure. We define the term "working directory" and explain the difference between absolute and relative path names. We also explain the functioning of distributed file systems and the operation of network directory services.

Systems Administration and Security [Learning Units 62 and 71]. The final set of our class lectures outlines the functions of systems administrators. We describe common tasks associated with these functions and introduce students briefly to systems management tools through a demonstration of some of the diagnostic capabilities available in the Windows 2000 operating system. In addition to explaining common systems administration functions, we emphasize that in smaller organizations, systems and network administrators not only may design the enterprise architecture, but may also be responsible for information assurance. We also introduce students to fundamental concepts associated with information assurance, including threats to physical security, data integrity, data confidentiality, and data accessibility. We outline a variety of security threats and common threat-mitigation and system-remediation techniques.

A Critique of Our Course

Our course, as currently taught, has strengths and weaknesses. Several students have provided the authors with unsolicited feedback acknowledging that concepts introduced in the class have helped improve their performance in their first programming class. This is anecdotal evidence at best and does not reflect a systematic survey of student perceptions or their performance in subsequent classes.

We have solicited feedback from several of the more advanced students who have taken the class. One student, taking the class in his senior year, provided the following comments concerning the course: "I love programming, and when I was in those classes I always wondered how the processor works. Even before I was in college, I wondered how RAM works, how data are stored on a hard-drive, the hardware-type questions one would ask. I think this class is a great class." An experienced network administrator taking the course found the content to be generally useful and appropriate. He noted that using a "systems thinking" approach was particularly useful and observed that many new employees lacked failed to demonstrate this type of analytic thinking.

However, the bulk of comments received in the class evaluations indicate that "too much information is presented for a two-credit-hour course" and that the class is too technical for beginning CIS students. Even the experienced network administrator cited above commented that some elements of the course were probably too technical. For example, he felt that the discussion regarding the differences between SRAM and DRAM technology probably was over the heads of students without a background in electronics.

Given our concern regarding the adequacy of our students' technical knowledge and comments received on course evaluations, we acknowledge that limiting this course to two-credit-hour course is not optimal. Furthermore, while students are provided a variety of simple programming assignments and other assignments calling for the use of Windows 2000 systems management

tools, the lack of laboratory facilities has prevented us from assigning structured practical exercises to complement more of the lecture material. We also note that our course as currently formulated does not present material identified in Learning units 68 and 69 concerning types of operating systems and the support of multimedia facilities. Recognizing these shortcomings, the CIS department is considering redesigning the course and/or repositioning it in the curriculum.

Conclusions

We agree with the specification of IS 2002 learning objectives and endorse the identified need to increase the technical orientation of IS graduates. However, our professional and academic experiences have alerted us to some concerns as to how well we can accomplish the objectives stated in IS 2002.4 while constrained by the 10-course curriculum limitation imposed by the accreditation standards for business schools (Gorgone, et al., 2002, pp. 7-8). As previously suggested, the current curriculum appears to require a significant tradeoff between the number of concepts that can be introduced and the level of detail in which these concepts can be presented. One of the guiding principles of the model curriculum report is that the effort should "represent a consensus from the IS community" (Gorgone, et al., 2002, p. 5). As we continue with evolving the design of our course, we would like to learn whether others within the IS academic community share our concerns and how they have addressed them in the design of their courses. Furthermore, we would like to see the IS community develop a consensus regarding a more detailed specification of IS 2002.4 course objectives as a means of assisting course and curriculum developers to best meet the needs of our students and their eventual employers.

References

Burd, S. D. *Systems Architecture*, Course Technology, Boston, MA, 2001.

Gorgone, J. T., Davis, G. B., Valacich, J. S., Topi, H., Feinstein, D. L., and Longenecker, H. E. J. **http://www.acm.org/education/curricula.html#IS2002**. Association for Computing Machinery (ACM), 2002. 3 March 2003.

Tanenbaum, A.S. *Structured Computer Organization*, Prentice Hall, Upper Saddle River, NJ, 1999.

Tanenbaum, A.S. *Modern Operating Systems*, Prentice Hall, Upper Saddle River, NJ, 2001.

Weill, P., and Broadbent, M. Leveraging the New Infrastructure: How Market Leaders Capitalize on Information Technology, Harvard Business School Press, Boston, MA, 1998.