

Experiences in Teaching an Online Course in Information Technology

**Scott F. Midkiff, Luiz A. DaSilva, William O. Plymale
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061 USA**

Abstract

Virginia Tech initiated a Master of Information Technology degree program in Fall 1998 and began online delivery of the program in Fall 2000. One of the first of three online courses offered in Fall 2000 was “Fundamentals of Computer Systems” which was developed and taught by the authors of this paper. The online version of the course was offered on a standard semester schedule and had due dates approximately every week, but was otherwise asynchronous. The initial and ongoing development and management of the course follows a model that is different than the traditional single-faculty instructor course model. There are five roles in this model: (1) course developers or content experts, (2) instructional designers, (2) course supervisor or “instructor of record,” (3) distance learning instructors, and (5) technical support personnel. In this paper, the authors share their experiences as course developers, course supervisor, and learning leaders for the “Fundamentals of Computer Systems” course. The paper discusses the course objectives, course design and delivery mechanisms, teaching and learning experiences, and “lessons learned” from the first offering of the course.

I. Introduction

To address workforce needs and to meet the growing demand for advanced course work in information technology (IT), Virginia Tech began a Master of Information Technology (MIT) degree and certificate program in 1998.^{1,2} The program is unique, at least for our university, in several key features.

- This single degree program is offered jointly by five departments in three colleges, specifically the Bradley Department of Electrical and Computer Engineering in the College of Engineering, the Department of Computer Science in the College of Arts and Sciences, and the Department of Accounting and Information Systems, the Department of Management Science and Information Technology and the Department of Management in the Pamplin College of Business. Representatives from these departments are members of a steering committee that sets policy, monitors course offerings, makes admission decisions, etc.
- The program format is modular. Students are required to complete 30 semester hours of course work, including four common foundation courses and three two-course modules in specialization areas such as networking, software development, and business information systems. As shown in Figure 1, one of the foundation courses is “Fundamentals of Computer Systems,” which is the subject of this paper.

Selection of Three Different Two-Course Modules					
Communications	Networking	Computer Engineering	Software Development	Business Information Systems	Decision Support Systems
Strategic Leadership					
Fundamentals of Computer Systems					
Software Engineering					
Object-Oriented Design with Java					

Figure 1. Master of Information Technology program format.

- Students can earn certificates as well as a degree. Students earn a certificate by completing any one of the two-course modules (e.g., Networking) together with an appropriate foundation course (e.g., Fundamentals of Computer Systems). For example, a student seeking knowledge in a specific area of IT can earn a certificate that indicates in-depth study in that specific area. Part-time degree students can earn certificates as “milestones” as they progress toward a degree.
- The program specifically targets non-traditional part-time students from a variety of undergraduate backgrounds. The initial geographic focus of the program was the northern Virginia area, home of a large number of IT companies and significant workforce needs, with the program offered at Virginia Tech’s Northern Virginia Center in Falls Church, Virginia. Beginning in Fall 2000, the MIT program is reaching out to students across Virginia using distance learning.

At the beginning of the Spring 2001 semester, there were 362 students enrolled in the MIT program, with 316 of these students taking the program as a mix of traditional and online classes at the Northern Virginia Center and 46 of these students enrolled strictly in the online format. Students from both of these categories took the “Fundamentals of Computer Systems” course that is the focus of this paper. Although the online program has been marketed only in Virginia, there are a small number of students enrolled in the program from four other states. Table 1 indicates the diversity of backgrounds of students enrolled in the program.

Table 1. Number of MIT Students Holding Each Type of Degree

	Bachelor of Science	Bachelor of Arts	Bachelor of Business Administration	Engineering Bachelors	Masters or Other Advanced Degree
Northern Virginia Center	236	60	8	12	56
Online	42	5	0	0	8

Given the desire to deliver the program to students across the Commonwealth of Virginia (and, potentially, beyond), online delivery of the program began in Fall 2000 with three courses being delivered in a mostly asynchronous format. There were three approaches available to taking the program state-wide: (i) web-based asynchronous delivery, (ii) synchronous delivery using streaming video carried by the Internet, and (iii) synchronous delivery using compressed video carried by Net.Work.Virginia³ using specialized distance learning classrooms. Synchronous delivery to specialized classrooms was not viable as existing state-wide resources are largely consumed by existing programs in engineering and business. Delivery using streaming video would need to be of limited quality. Perhaps more importantly, synchronous delivery does not provide the scalability desired for the MIT program and it does not provide the flexibility sought by students, many of whom travel frequently as part of their jobs. For these reasons, the program is using asynchronous web-based delivery.

One of the online courses offered in Fall 2000 was “Fundamentals of Computer Systems” which was developed and taught by the authors. In this paper, we describe the course and relate our experiences in teaching the class this first time. In Section II, we describe the functional roles used to design, develop, and implement this online course. In Section III, we describe the course and, in Section IV, its asynchronous online format and the instructional technology components employed in delivery. A number of lessons were learned and these are shared in Section V.

II. Design, Development, and Delivery

The “Fundamentals of Computer Systems” course had been taught twice prior to Fall 2000 using a traditional classroom lecture format. There are several reasons why this traditional model of a single instructor being completely responsible for the design and delivery of the course could not meet our requirements for online delivery.

- Transforming a course to an asynchronous online format usually requires a significant development effort and can consume a substantial amount of time for support personnel. This development effort must be amortized over multiple offerings of the class with, presumably, different instructors.
- Few people have all of the skills and knowledge (not to mention time) needed to design a course, realize it as a web-based course, and teach the course.
- The traditional approach is not scaleable. There could, potentially, be hundreds of students in a class during a term. This would consume too much of a single instructor’s time (assuming traditional faculty or part-time instructors).
- The traditional model does not provide an inherent process for content and quality control. Such control is particularly critical if the online version is taught by part-time or adjunct instructors. In fact, instructors for the course could be located anywhere as the Internet-based nature of the course removes geographical barriers for instructors as well as for students.

To overcome the limitations of the traditional approach, we employed a process with five functional roles.

1. Content expert
2. Instructional design
3. Instructor of record
4. Distance learning instructor
5. Technical support

One or more content experts are required to design the course syllabus, prepare lecture content, and develop at least some exercises. The content experts generate material in some format that can be transformed to an appropriate format for actual delivery. The content experts need not be proficient with a specific web-based delivery or course management system. In our own case, one content expert (Midkiff) developed the course syllabus and two content experts (Midkiff and DaSilva) prepared “lectures” and a list of learning objectives, a list of review topics, and a short quiz for each lecture. The lectures, delivered as audio content with synchronized graphics (see Section IV) were delivered to instructional design specialists as Microsoft PowerPoint files and audio recordings on digital mini-discs. The bulk of the content experts’ effort is expended prior to delivery. However, some later effort is required to improve and update content. A content expert may never actually teach the course, although such experience would be beneficial.

Instructional designers create the overall structure of the class web site and transform the content provided by content experts to formats needed for actual delivery. For example, a Microsoft PowerPoint file and audio recording are compressed and synchronized for streaming delivery. The RealNetworks RealPlayer⁴ format was used for lecture content for our class. Microsoft Word files are transformed to BlackBoard⁵ quizzes or HyperText Markup Language (HTML) files for lists of review topics and learning objectives. The instructional designer is also responsible for user interfaces issues, including usability and design for accessibility. For the online MIT program, Virginia Tech’s Institute for Distance and Distributed Learning (IDDL)⁶ provides the instructional design function. The instructional designers’ greatest effort is to prepare for the first delivery of the course. However, instructional designers must also assist with updates and, during a course offering, with web site maintenance.

The instructor of record (IOR) is responsible for overseeing each offering of the course and for evaluating student work – probably with grading assistance – and assigning grades. The IOR is presumably a faculty member with ongoing overall responsibility for the course. The IOR need not be a content developer or an instructor, although serving in such roles on occasion will likely be beneficial. For our course, one faculty member (Midkiff) served as a content expert, IOR, and a distance learning instructor.

A distance learning instructor (DLI) actually runs a section of the class when it is offered. The DLI is responsible for interacting with the students in his or her section. For our class, this interaction was primarily through threaded discussion lists, interactive chat sessions, and electronic mail (see Section IV). With the cooperation of the IOR, the DLI may also be responsible for generating homework, project, and other assignments. The number of DLIs required depends on the desired section size which, in turn, depends on the amount of interaction and time commitment for a DLI. For classes that are largely discussion based, small section sizes would be desirable. For our course, we originally decided to have up to 35 students in each

of two sections, each led by a separate DLI (Midkiff and Plymale). We actually had only 31 students in two sections for an average section size of less than 16. This section size was manageable for the amount of interactivity required by the class. However, given the level of help needed due to the diverse backgrounds of the students, our initial target of 35 students per section was probably unreasonably high.

The final functional role is that of technical support. Technical support staff are needed during each offering of the class, primarily, to maintain web servers and other generic resources used by instructors and students. Support staff are also needed to assist with interfacing to administrative systems, e.g., for student enrollment. The IDDL and Educational Technologies Department⁷ at Virginia Tech provided support for a general course web server and enrollment and for Blackboard CourseInfo course management system, respectively.

While they may need to be refined over time and for courses that differ substantially, we found the functional roles and design, development, and delivery process to be effective. Of course, strong interfaces are needed between the five functional roles. In our case, this was facilitated by extremely cooperative and professional support organizations and by having a single faculty member serve in the roles of content expert, IOR, and DLI for this first offering. Where the content expert is not serving as a DLI, there needs to be some feedback mechanism to guide the content developer in updating content.

III. Fundamentals of Computer Systems Course

As noted in Section I, the “Fundamentals of Computer Systems” course is required for all MIT program students. The course is highly technical compared to most “information technology” courses. It provides students with the fundamental knowledge of information representation, digital hardware, computer organization, and data networks needed for broad knowledge in the IT arena and, for interested students, to proceed to the MIT program’s Networking or Computer Engineering modules.

Students taking this course can be briefly characterized as follows.

- Students are located throughout Virginia with some, potentially, outside of Virginia.
- Students are employed full-time while taking the class.
- The students are mature. They have an undergraduate degree, but perhaps in a non-technical field.
- Many, but not all, students have some work experience in the IT industry that is related to the class.
- Since this is a foundation course, it is the first class in the MIT program for many of the students.
- For many students, several years have passed since taking their last academic, for-credit class.
- Students have at least one high-level language programming course as prerequisite background.
- Since they are working full-time and have other demands on their time, students value their time and need to see the value of time that they spent on the class.
- Students want flexibility, specifically “any time, any where” access.

- Students want to access content from home or while traveling with a standard 56 kilobit per second modem or from work behind a corporate firewall.

The course discusses the fundamental principles and concepts of computer systems including Boolean logic, number systems and information representation, design and operation of combinational and simple sequential digital logic, basic computer organization and instruction set architectures, interaction of the operating system and hardware, and networks and data communication. Specific learning objectives are provided in Figure 2. Students are expected to have had some experience in programming and computer use prior to taking this course. Students should understand how to program in a modern high-level language such as C, C++, or Java and should know Microsoft Windows fundamentals to use online resources, a text editor, and course software. The course partially duplicates existing computer organization and digital design courses taken by computer engineering, electrical engineering, and computer science majors. MIT students that recently earned undergraduate degrees in those disciplines are exempted from taking the course.

- | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Having successfully completed this course, a student should be able to:</p> <ul style="list-style-type: none"> • Express and derive logical operations using Boolean operators. • Design and analyze combinational and synchronous sequential logic circuits. • Express and analyze sequential behavior using timing diagrams. • Represent numerical and non-numerical data using standard encoding methods. • Determine the content of data packets given packet formats. • Explain the basic organization of a computer, including functions of the central processing unit, cache memory, main memory, mass storage, and other input/output devices. • Explain the principles of operation of von Neuman architectures. • Explain basic performance improvement schemes for computers, including cache, pipelining, and parallel processing. • Analyze instruction formats and explain the fetch, decoding, and execution of standard instructions. • Explain standard operating system functions and the relationship between system hardware, operating system software, and software applications. • Explain the basic operation of network protocols and systems for data communications. |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Figure 2. Course learning objectives.

In addition to online resources, students have a text book and utilize two software packages. The text is *Principles of Computer Architecture* by M. J. Murdocca and V. P. Heuring⁸. This is a self-contained and readable text, making it suitable for a distance learning course. A logic simulator, LogicWorks 4⁹ from Capilano Computing Systems, is purchased by students to support the logic design portions of the course. LogicWorks runs on both Microsoft Windows and Apple Macintosh platforms. The ARCTools instruction set simulator for the example ARC (A RISC Computer) processor used in the text is used to support computer organization and assembly language programming portions of the course. Students download this simulator from the textbook authors' web site. Since it is written in Java, ARCTools runs on any platform, including Microsoft Windows and Apple Macintosh, that supports the Java Run-Time Environment.

There are three types of graded assignments: (i) homework, (ii) projects, and (iii) exams. In Fall 2000, there were nine homework assignments consisting of problems from the text and problems

developed by the instructors. There were four design projects, with two projects focusing on logic design and two projects involving computer organization and assembly language programming. Projects are completed using the logic simulator and instruction set simulator and students submit written reports and simulation or source code files. There were two exams, a mid-term and a final. Both exams were open book “take-home” exams. In addition to graded assignments, students are expected to complete reading assignments in the textbook, view online “lectures,” complete non-credit self-assessment quizzes, and participate in online discussions and chat sessions.

IV. Asynchronous Web-Based Delivery

The asynchronous web-based version of the course was designed to meet the needs of the target audience, as described in Section III, maintain the required rigor for the course, and require reasonable effort for development on the part of the content experts, instructional designers, and technical support staff. The course is organized as depicted in Figure 3.

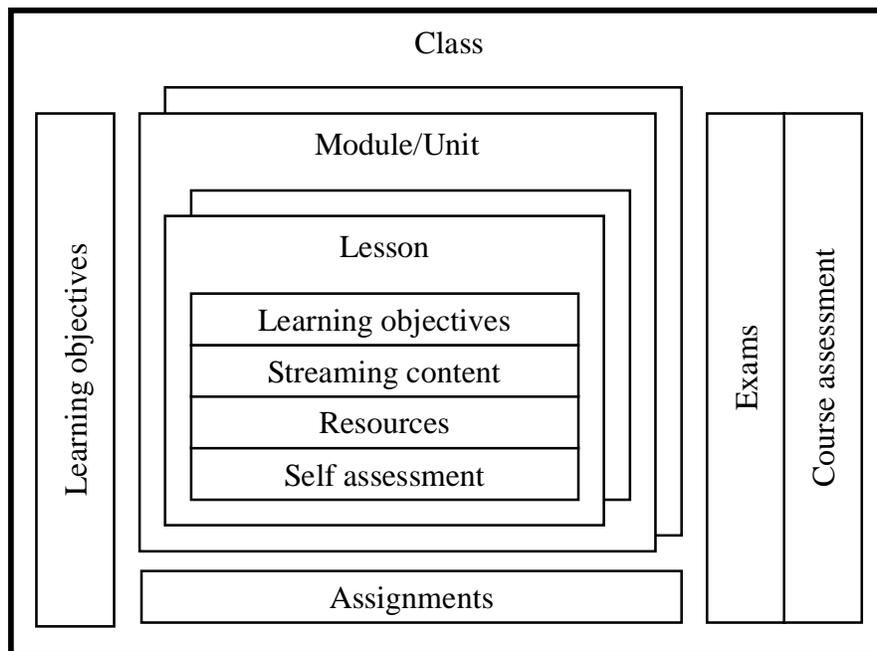


Figure 3. Structure of the web-based course.

The course content is decomposed into eight different modules which represented the course’s major topic areas. Each module is further decomposed into one or more units. The outline of the modules and units is shown in Figure 4. In aggregate, the modules provide content to meet the course’s learning objectives, as depicted in Figure 3. A take-home midterm and final exam are used to evaluate a student’s mastery of the course material. A course evaluation instrument was used to survey the students at the end of the course. Ideally, formative evaluation would be used throughout the semester to provide improvements during a single offering of the course. While we plan to do this for future offerings, it was not done for the Fall 2000 offering of the course.

<p>Module I: Introduction (3 lessons)</p> <p>Unit A: Course Overview</p> <p>Unit B: Computer Systems</p>
<p>Module II: Data Representation (7 lessons)</p> <p>Unit A: Fixed Point Number</p> <p>Unit B: Floating Point Numbers</p> <p>Unit C: Other Codes</p>
<p>Module III: Digital Logic (18 lessons)</p> <p>Unit A: Combinational Logic</p> <p>Unit B: Combinational Logic Building Blocks</p> <p>Unit C: Sequential Logic</p> <p>Unit D: Finite State Machines</p>
<p>Module IV: Instruction Set Architecture (12 lessons)</p> <p>Unit A: Hardware Components of the Instruction Set Architecture</p> <p>Unit B: ARC: Example RISC Computer</p> <p>Unit C: Assembly Language Programming</p> <p>Unit D: Examples of Other Processors</p>
<p>Module V: Datapath and Control (8 lessons)</p> <p>Unit A: Microarchitecture for the ARC</p> <p>Unit B: Hardwired Control</p>
<p>Module VI: Memory (5 lessons)</p> <p>Unit A: Memory Devices</p> <p>Unit B: Memory Systems</p>
<p>Module VII: Input and Output (5 lessons)</p> <p>Unit A: Input and Output Systems</p> <p>Unit B: Input and Output Devices</p>
<p>Module VIII: Data Communication and Networks (11 lessons)</p> <p>Unit A: Plain Old Telephone Service and Data Networks</p> <p>Unit B: Network Architectures</p> <p>Unit C: Network Technologies</p>

Figure 4. Course outline organized as modules and units.

There are a total of 69 lessons. Each lesson consists of a self-contained unit with its own specific learning objectives, as shown in Figure 3. The primary content of each lesson is a single 5- to 20-minute online lecture. We tried to keep the online lectures short enough to enable flexible scheduling and to hold the students' attention. The online lectures use streaming audio and are synchronized with graphics using RealNetworks RealPlayer format. For convenience, the streaming content is also available in a format that students can download and play later without being connected to the Internet. These could be made available to students on a CD-ROM, but were not for the first offering of the course since content development was ongoing throughout the semester. The graphics are presentation slides that are available to students as Adobe Acrobat¹¹ Portable Document Format (PDF) files. As shown in Figure 3, each lesson also includes reading assignments, a set of review topics and other resources that support the lesson, and an online non-credit self-assessment quiz that students can use to determine if they have mastered the lesson's learning objectives.

The topical decomposition of the course content into modules, units, and lessons is mapped onto a standard 15-week academic semester. A one-week block is used as the basic unit of time. A set of lessons and one assignment – a homework, project, or exam – are assigned during each week. While an assignment is due every week, students nominally have a two-week window to complete a given assignment. This provides students with scheduling flexibility.

The preferred modes for interaction are threaded discussion lists (forums) and synchronous “chat” sessions. Some questions are submitted by electronic mail, but these are supposed to be limited to questions that revealed solution information or that concerned grades or other personal matters.

Discussion forums are created for each module, for questions about lessons in a module, and for each assignment, for questions and updates on an assignment. Students are allowed to post to all forums except for the exam forums. There are separate discussion lists for each section of the class (two in Fall 2000). While students occasionally answered questions from other students, the most common postings to the forums in Fall 2000 were questions by students, answers from the distance learning instructor, and updates or hints from the instructor.

Optional chat sessions are held approximately every week. In Fall 2000, one common session was usually held for both sections. Each chat session had a primary topic, usually a particular project assignment. The chat format was strictly text-based. Using Blackboard’s CourseInfo chat facility, students can access a general chat area and also submit questions using a “Q&A” area of the chat applet. The instructor responds to the questions. Questions and corresponding answers are then made visible to all students. A transcript of the question and answer section of the chat session is captured and made available to students.

Homework, project, and exam assignments are posted as either HTML or PDF files. In Fall 2000, students were allowed to submit most assignments using a web-based “digital dropbox” function built into Blackboard’s CourseInfo, electronic mail, or fax or standard mail (for written work). The digital dropbox is, by far, the most convenient method for collecting work. Assignments are archived by the system and are accessible to the instructor and to teaching assistants. However, some students had problems with accessing the digital dropbox and other alternative schemes were provided in the Fall 2000 offering of the class.

The forums, chat sessions, online self-assessment quizzes, digital dropbox, and an online grade book are all provided by Blackboard’s CourseInfo version 3.11. Students can access these features from a standard web browser. CourseInfo itself ran on a Sun Enterprise 3500 server and supported numerous other classes. Other course content was provided on a standard web server. Links provided access to CourseInfo components.

CourseInfo offers a group feature that allows the segmenting of students into smaller study groups. CourseInfo groups consist of private areas that students and DLIs can use for discussions, chats, file sharing, and electronic mail access. Access to each of these group areas can be controlled by the instructor. Several two-person groups worked on two of the projects and two students used CourseInfo’s group feature. Several group members expressed

appreciation for project assistance from their partners. Group projects also reduced grading time. While individual group members were required to write their own project reports, sections of the projects, e.g., simulations, were similar.

IV. Lessons Learned

While the Fall 2000 offering of “Fundamentals of Computer Systems” was generally very successful, we did gain experience and have developed a list of “lessons learned” and “wish list” items that will be used to improve this online course and to guide the development of other courses in Virginia Tech’s online MIT program. The key lessons learned are provided below.

- Content should be developed early. We were still developing content during the semester, along with trying to perform other duties. Content development is very time consuming, even with a strong instructional design support team. It should be spread across time and, preferably, across people. The multiple roles defined in Section II allowed us to divide the content development task among two people (Midkiff and DaSilva).
- The development time for content experts is high. It is a superset of preparation time for a synchronous video class, but for fewer hours of content. The time demand for a distance learning instructors can also be high. Time is spent responding to forum postings and electronic mail and participating in chat sessions, but there is no time required for lecture or preparation.
- New teaching credit models are needed for the new roles of content developer, instructor of record, and distance learning instructor.
- The logistical tasks at the beginning of the semester are particularly challenging for an online course, especially where this is the first such course for many of the students. Access to the course content and electronic mail addresses are based on Virginia Tech personal identifiers (PIDs). While this system is simple, it works only if all students know their PID, set a password, and check their Virginia Tech (*pid@vt.edu*) electronic mail accounts. Better information is to be provided as part of the enrollment process to address this problem.
- An associated “start-up” problem is associating course management system enrollments with actual course enrollments. We employed a self-registration scheme which, we thought, would be simpler. However, students had problems with this and the system was not robust when errors were made. Using an automated link between actual course registration and course management system registration would likely be better.
- The overall structure of the web site can be simplified. The original design had separate sites for the overall course and for each section. A unique site is needed for each section (with many common documents), but a common course site is not needed. This will simplify access by students, instructors, and instructional designers.
- We found the Blackboard CourseInfo version 3.11 forum to be very effective. However, the chat tool is lacking many features, including easy ways to display graphics and capture

transcripts. The “Q&A” paradigm is effective and provides a good way to manage active chat sessions. Educational Technologies is planning an upgrade to Blackboard version 5 with an anticipated production release in Summer 2001. This upgrade should address several problems including transcripts of chat sessions.

- Using the digital drop box for submissions is much more manageable than using electronic mail attachments, fax, or standard mail. However, it is not reasonable to ask students to prepare electronic submissions for many assignments. Possible solutions include scanning documents and designing assignments that rely more on simulations, document templates, or other methods where electronic content is available.
- Proctored exams are needed to ensure academic integrity. The IDDL has developed a scheme for approving proctors that is manageable, but it does require extra effort on the part of both students and the instructor of record. It is also counter to the asynchronous, self-pace approach of a web-based course.
- More effective methods are needed for student interaction for some matters. While students are mostly very positive about the course and the online delivery, there is a need for direct dialog (telephone or face-to-face meetings) to deal with some questions and problems, especially in a technical course such as this one. The greatest difficulty was in the use of course software.
- Methods are needed to motivate increased student participation in the forums and chat sessions. Students need to assist each other more. Some possible approaches include use of study groups, unsupervised chat sessions, threaded discussion directed at exchange of ideas among students, and group projects (which was shown to be successful on a limited basis).
- Better feedback is needed to guide course design. Due to the distance between content experts and students, it is not easy to determine what works and what does not, which points require more complete explanation or examples, etc. Formative evaluations are needed throughout the semester, e.g., that are associated with each module, to provide feedback to improve course content and delivery.
- It is difficult to effectively utilize a graduate assistant to assist the content expert. To get meaningful contribution from a graduate assistant in developing content requires significant advance planning and supervision on the part of the content expert. If these are there, the assistant may be able to help with some of the time-consuming tasks such as preparing graphics for the slides, creating the self-assessment instruments, etc.
- It is important to properly set expectations of both students and instructors prior to the start of the course. Instructors (faculty) usually believe that online instruction is more demanding and less rewarding than traditional classroom instruction. Student must have reasonable expectations about how to communicate with instructors, how much help can be provided, and response times. Students must take more responsibility for their own learning.

- The distance learning instructor role is new and expectations must be defined. This is especially important since the DLI is likely working full-time at another job or has other demanding duties. The role of the DLI versus the content expert, instructor of record, and instructional designer for managing and maintaining the course must be clearly defined. Students and the DLI may perceive (incorrectly) that he or she should be available around-the-clock, seven-days-a-week. Methods are needed to determine the amount of time a potential DLI can allocate to a course and to determine a potential DLI's knowledge of course subject matter. Weakness in subject areas can result in slower response times, incorrect responses to student questions, and ineffective interactions in the forum and chat sessions.

We are pursuing solutions to these problems for future offerings of this course and for other courses in the online MIT program. Our technical support staff and instruction designers are working to make the latest course management tools available. We are seeking and investigating methods to encourage more student participation and to better define expectations. We will be using a proctored final exam for future offerings. And, we will add mechanisms for student feedback at the module level to improve course content and delivery.

V. Conclusions

This paper described the asynchronous web-based offering of "Fundamentals of Computer Systems," a foundation course in Virginia Tech's Master of Information Technology degree and certificate program. This relatively technical course provides IT students with fundamental knowledge of digital hardware, computer systems, and data communications. The online version of the course is built around a number of small lessons that consist primarily of streaming audio with synchronized graphics. Chat sessions and a threaded discussion list are used to facilitate interaction. Homework, projects, and exams are used to reinforce key concepts and to evaluate students' mastery of learning objectives.

The online version of the course was generally well received by the 31 students enrolled in the Fall 2000 offering of the course. Changes are needed, however, to improve learning, to scale the course to larger enrollments, and to reduce time demands on content experts and instructors.

Acknowledgements

This work was supported in part by a grant from Virginia Tech's Center for Innovation in Learning and, through related projects, by the Southeastern University and College Coalition for Engineering Education (SUCCEED), a National Science Foundation Engineering Education Coalition (Cooperative Agreement 9727411). The efforts of Virginia Tech's Institute for Distance and Distributed Learning, especially Miriam Guthrie and Jonathan Pratt, and Virginia Tech's Educational Technologies Department are gratefully acknowledged.

Bibliography

1. W. H. Tranter and L. A. Ferrari, "The Graduate Program in Information Technology at Virginia Tech," *Frontiers in Education Conference*, November 10-13, 1999, San Juan, P.R., pp. 12b5.1-12b5.5. (Available at <http://fie.engrng.pitt.edu/fie99/>.)
2. URL: <http://www.conted.vt.edu/it@vt.edu>.
3. URL: <http://www.networkvirginia.net>.
4. URL: <http://www.real.com>.
5. URL: <http://www.blackboard.com>.
6. URL: <http://www.iddl.vt.edu>.
7. URL: <http://www.edtech.vt.edu>.
8. M. J. Murdocca and V. P. Heuring, *Principles of Computer Architecture*, Prentice-Hall, Upper Saddle River, NJ, 2000.
9. Capilano Computing Systems, LogicWorks 4 (software), Prentice-Hall, Upper Saddle River, NJ, 1999.
10. URL: <ftp://schof.colorado.edu/pub/TTLSystem/ARCTools/>.
11. URL: <http://www.adobe.com/acrobat>.

SCOTT F. MIDKIFF

Scott Midkiff is a Professor in the Bradley Department of Electrical and Computer Engineering at Virginia Tech. He teaches courses in computer networks and computer engineering and is a member of the Master of Information Technology program's steering committee. He conducts research in computer networking. Midkiff received the Ph.D. in 1985 from Duke University. He is a member of the ASEE, IEEE, and ACM.

LUIZ A. DASILVA

Luiz DaSilva is an Assistant Professor in the Bradley Department of Electrical and Computer Engineering at Virginia Tech. He teaches computer engineering and in telecommunications courses and has been active in distance and distributed learning for the past two years. Dr. DaSilva conducts research in computer networks and mobile systems. He received a Ph.D. from The University of Kansas in 1998. He is a member of the ASEE and IEEE.

WILLIAM O. PLYMALE

Bill Plymale is the Instructional Technologies Systems Manager in the Educational Technologies Department at Virginia Tech. His work areas have included the Learning Resource Center's Research and Measurement Division, numerous Information Systems initiatives, and EDP audit manager in the Department of Internal Audit. He holds a B.S. in Psychology, a B.S. in Computer Science, and an M.S. in Electrical Engineering, all from Virginia Tech.