Real-Time Specialization Track
Masters of Software Engineering (MSE) Program

Carol L. Hoover
March 1, 1995
CMU-CS-95-122

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890

This report is a revision of an internal report written and distributed within the CMU community by Carol L. Hoover in August, 1994.

Author's Internet Address: clh@cs.cmu.edu

Abstract

There is an increasing demand for software to support real-time applications such as air traffic control, factory automation, patient monitoring systems, and video-conferencing. Software engineers who develop real-time software need to understand functional constraints such as timing, fault tolerance, and reliability that distinquish real-time from non-real-time applications. The Real-Time Specialization Track within the Masters of Software Engineering (MSE) Program at Carnegie Mellon University enables graduate software engineering students to acquire knowledge about real-time applications, to obtain an understanding of fundamental principles of real-time computing, and to gain experience in the development of real-time software. The track is flexible in that it allows students to choose their own set of track courses from a list of approved real-time courses. This report describes the track rationale and requirements with educational guidelines to help students select courses, lists the approved courses, and presents brief synopses of the approved courses. The track is based on the author's work in developing specialization tracks for the MSE Program. The author started this work during the spring of 1993.

Table of Contents

1 Rationale

Masters of Software Engineering (MSE) students who pursue the *Real-Time Specialization Track* will develop a broad view of the role of the software engineer in real-time software development. They will be able to identify common characteristics of real-time applications and to distinguish between applications which require real-time solutions and those which only require adequately fast response times. In addition, they will be aware that real-time applications often involve reliability, safety, and fault tolerance requirements.

Students will have the opportunity to acquire knowledge and skills related to topics, such as the following, which are important in the development of real-time software and systems.

- Specification of requirements and design.
- Design, implementation, and evaluation of real-time software.
- Real-time scheduling theory.
- Real-time software architectures/impact of timing, synchronization, and fault tolerance on the design of real-time software architectures.
- Languages used to develop real-time software.
- Real-time operating systems.
- Hardware/software interfaces and integration.
- Methods for analyzing real-time system performance.
- Design and implementation of reusable software components for real-time systems.
- Quantitative analysis and design of high-performance, real-time computing systems.
- Operating system support for real-time applications such as multimedia.

Researchers at Carnegie Mellon University develop state-of-the-art multimedia technology and teach courses to help transfer their ideas to students. Students interested in multimedia technology and applications can select courses to learn about:

- Media representation, storage, communication, and digital processing of audio, still images, and video.
- State-of-the-art techniques used to develop real-time operating system and multimedia applications software.

Students should note that topics covered in multimedia courses can often be applied to other real-time applications.

2 Track Requirements

Students pursuing the Real-Time Specialization Track will complete **four or more full-semester (9-12 unit) courses including** *Introduction to Real-Time Software and Systems* (15-679). Students will start the track by taking this course. They will take more specialized courses in subsequent semesters. The reader should refer to *Section 3: Approved Courses* for the list of approved courses and to *Section 4: Course Descriptions* for brief course descriptions. Students who want to take courses which are not on the approved list to satisfy the four-course requirement must seek his/her advisor's permission.

Two half-semester or two 6-unit courses count for one full-semester course. *Real-Time Software Design* (17-612) and *Advanced Topics in PS: Operating System Support for Real-Time Systems* (15-810) are 6-unit courses. Completion of both of these courses is equivalent to the completion of one full-semester course. In addition, a student may pursue an independent study which is equivalent to one half-semester course (6 units), one full-semester course (9-12 units), or one full-semester course plus one half-semester course (15-18 units).

Students should select approved courses that correspond to their educational objectives. *Table 1: Mapping Educational Objectives to Approved Courses* is a guide for selecting appropriate courses.

Table 1: Mapping Educational Objectives to Approved Courses

Educational Objective	Corresponding Approved Course
Acquire fundamental knowledge and skills related to real-time computing and real-time software development.	Introduction to Real-Time Software and Systems (15-679)
Gain experience in the design and implementation of real-time software.	Real-Time Software Design (17-612)
Apply state-of-the-art techniques to the development of real-time operating system and multimedia applications software.	Special Topics in Computer Science: Multimedia Systems (15-499X)
Acquire knowledge about the design and implementation of operating systems which support real-time applications and develop the ability to summarize and critique literature in the field.	Advanced Topics in PS: Operating System Support for Real-Time Systems (15-810X)
Acquire knowledge about media representation, storage, communication, and digital processing of audio, still images, and video.	Media Technology (15-820X)
Acquire knowledge and/or skills in an individually selected area of real-time computing.	Independent Study of Real-Time Computing Issues (17-799)

Table 1: Mapping Educational Objectives to Approved Courses

Educational Objective	Corresponding Approved Course
Acquire knowledge and skills needed to analyze the performance of real-time system components and devices.	High Performance Computing Systems (18-549)
Acquire knowledge needed to develop software for digital signal processing applications.	Digital Signal Processing I (18-791)

The prerequisite for *Digital Signal Processing I* (18-791) is *Signals and Systems* (18-396), an undergraduate Electrical and Computer Engineering course. Students who have already taken an undergraduate signals and systems course do not need to take *Signals and Systems* before taking *Digital Signal Processing I*. Students should also note that the course *High Performance Computing Systems* (18-549) involves the performance analysis of system components used in real-time applications, some but not all of which involve multimedia technology.

3 Approved Courses

The student will select the equivalent of 4 or more full-semester (9-12 unit) courses from those listed in sections 3.1 and 3.2 including *Introduction to Real-Time Software and Systems* (15-679).

3.1 School of Computer Science

15-679 12 units Fall	*Introduction to Real-Time Software and Systems* Instructor: Carol Hoover Required Course	
17-612 6 units Spring	*Real-Time Software Design* Instructor: Jorge Diaz-Herrera	
15-499(X) 12 units Spring	*Special Topics in Computer Science: Multimedia Systems* Instructor: Ragunathan Rajkumar	
15-810 6 units Fall/ Spring/ Summer	*Advanced Topics in PS: Operating System Support for Real-Time Systems* Instructor: E.N. Elnozahy Contact the instructor to determine if the course is being offered during a particular semester.	
15-820(X) 12 units Spring	*Media Technology* Instructor: Roger B. Dannenberg	
17-799 6-18 units Fall/ Spring/ Summer	*Independent Study of Real-Time Computing Issues* Advisors: Individually arranged. The student and his/her independent study advisor will determine the topic to be studied and/or project to be completed.	

3.2 Department of Electrical and Computer Engineering

18-549 12 units Spring	*High Performance Computing Systems* Instructor: Jay Strosnider
18-791 12 units Fall	*Digital Signal Processing I* Instructor: Virginia Stonick

4 Course Descriptions

4.1 Introduction to Real-Time Software and Systems (15-679)

The primary purpose of this course is to present an overview of real-time computing. Basic concepts, terminology, and problems of real-time computing are introduced. The constraints of real-time computing are used to contrast real-time applications from applications that are not real-time. The course focuses on software solutions to real-time problems. Issues that are addressed include scheduling, specification of system requirements and design, real-time software architectures, languages and operating systems for real-time computing, real-time problems in a distributed processing system, and hardware-software interfaces. After successfully completing this course, the student will be able to identify and propose solutions to basic problems in real-time computing. It is the goal of this course to motivate and prepare the student to pursue a more in-depth study of specific problems in real-time computing.

Prerequisites: Proficiency in one high-level programming language used to develop real-time software (e.g. C, C++, or Ada). Knowledge of basic operating system concepts generally taught in an undergraduate operating system course.

4.2 Real-Time Software Design (17-612)

This course teaches engineering principles and construction techniques that are particularly applicable to the design and development of real-time software systems. The course will address specific timing, synchronization, and fault tolerance issues that have an impact on the system architecture and software structure of real-time software systems. The course touches on five major topic areas: real-time requirements, external interfaces, software design methods, time and fault control, and concurrent programming. The course focuses on practical applications implemented using Ada-based technology. Prototyping and other computer assisted design aids are an integral part of the laboratory exercises. A particular application of computer real-time control serves as the major component of the laboratory materials.

Prerequisites: Introduction to Real-Time Software and Systems (15-679) or equivalent background. Working knowledge of a modern high-level programming language, basic software design concepts, and software architecture notions.

4.3 Special Topics in Computer Science: Topics in Multimedia Systems (15-499X)

The primary objective of this course is to learn hands-on programming skills in systems support for the fast-growing field of real-time multimedia applications. This course takes a project-based approach to the design and development of multimedia systems with an emphasis on current research in real-time and multimedia operating system issues. Projects will involve developing multimedia applications and new kernel mechanisms to support this exciting field of emerging applications. The implementation environment for these projects will be the Real-Time Mach microkernel and Unix programming environment developed at CMU.

Topics to be discussed include: multimedia applications, Mach, Real-Time Mach, systems support for multimedia, multimedia communications and networking, real-time systems, and video compression/decompression. Some past group projects include a Quality-Of-Service (QOS) server for Real-Time Mach, real-time audio multiplexing, real-time digital effects, enhanced multimedia schedulers, adaptive virtual memory management, and guaranteed real-time traffic on ATM.

Prerequisites: Operating Systems (15-412) and Discrete Mathematics (21-228), or by instructor permission.

4.4 Advanced Topics in PS: Operating System Support for Real-Time Systems (15-810)

The focus of this class is on operating system techniques to meet the demands of multimedia/real-time applications. Topics to be covered include implementation/design techniques rather than application-level issues of multimedia/real-time systems. In each session, a student will present a paper by summarizing it. This will be followed by a discussion of the main points addressed in the paper. Each student will also be asked to write referee-style reports for three papers discussed in class over the semester. This is done to encourage active participation and to teach students how to write quality referee reports, an essential activity that any researcher is expected to accomplish throughout his or her career.

4.5 Media Technology (15-820X)

This course teaches the fundamentals of media representation, storage, communication, and processing by digital means, with an emphasis on audio, still images, and video media. It begins with an introduction to sampling theory and various representation techniques. This is used to describe and explain a variety of real devices, formats, and standards. Students will learn to analyze media technology in terms of critical properties such as resolution, noise, bandwidth, latency, and computation.

Topics to be covered include: audio sampling theory and processing, 2D image sampling, color spaces, audio compression, image compression, video compression, storage devices including CD-ROM, RDAT, and holographic storage, storage formats including CD-I and HyTime, time codes and synchronization, communication technology including ATM, ISDN, and cellular radio, input technology, and application examples.

4.6 High Performance Computing Systems (18-549)

The course focuses on quantitative analysis and design of high-performance, application-specific computing systems. System specifications will first be developed, then scheduling theoretic techniques will be used to quantitatively explore the design space. Performance models will be developed to expose the hardware/software boundary issues for each of the following subsystems: CPUs, backplane buses, caches, disk subsystems, digital signal processors and data capturing subsystems. Composite hardware/software system level models will then be developed to optimize system performance.

Prerequisite: Fundamentals of Computer Engineering (21-228) or equivalent knowledge.

4.7 Digital Signal Processing I (18-791)

Principles of discrete-time signal processing. Topics include discrete-time Fourier transforms, Z-transforms, discrete Fourier transforms, fast Fourier transform algorithms, digital filter design and implementation techniques. In addition, the course includes an introduction to the effects of finite register length, multivariate and multi-dimensional signal processing, with applications to speech and image processing.

Prerequisite: Signals and Systems (18-396) or equivalent knowledge.

Acknowledgments

The author’s work in specifying the Real-Time Specialization Track began with the development of the TAP-D Model for developing specialization tracks in the MSE Program [Hoover93]. Using the model, she outlined a set of knowledge bases and mapped the knowledge components to educational objectives. With the contributions of several faculty members and researchers, she then selected a set of graduate courses offered at Carnegie Mellon University whose educational objectives are identical to or similar to our target objectives.

The author acknowledges with appreciation the following people for their advice regarding the real-time-related courses that they teach: Roger Dannenberg, Jorge Diaz-Herrera, E.N. Elnozahy, Ragunathan Rajkumar, Virginia Stonick, Jay Strosnider, and Don Thomas.

She also thanks Roger Dannenberg, Nancy Mead, Cliff Mercer, and Mary Shaw for their help in specifying the content of Introduction to Real-Time Software (17-880), a half-semester graduate course which was the forerunner of Real-Time Software and Systems (15-679), a full-semester graduate course.

The author thanks the members of the MSE Executive Committee (David Garlan, Phyllis Lewis, Nancy Mead, Mary Shaw, and James Tomayko) for their review of this track. She thanks Mary Shaw for her ideas about specialization tracks. She recognizes the commitment of the members of the Real-Time Track Team (Jorge Diaz-Herrera, Carol Hoover, Phyllis Lewis, Ragunathan Rajkumar, Ellen Saxon, and Lui Sha).

References

[CIT94] Carnegie Institute of Technology. *Electrical and Computer Engineering. Research and Graduate Studies*. Pittsburgh, Pa.: Carnegie Mellon University, 1994.

[CMU94] Carnegie Mellon University. *Undergraduate Catalogue*. Pittsburgh, Pa.: Carnegie Mellon University, 1994.

[Hoover93] Hoover, Carol L. *TAP-D: A Model for Developing Specialization Tracks in Graduate Software Engineering Education*. Technical Report CMU-CS-93-181, Pittsburgh, Pa.: Carnegie Mellon University, 1993.

[MSE94] Master of Software Engineering Program. *MSE Program Description*. Pittsburgh, Pa.: Carnegie Mellon University, 1994.

[SCS94] School of Computer Science. *Graduate Studies in Computer Science*. Pittsburgh, Pa.: Carnegie Mellon University, 1994.