R·I·T

B. Thomas Golisano College of Computing and Information Sciences
Department of Software Engineering

NEW (or REVISED) COURSE: 4010-549

1.0 **Title**: Seminar Topics in Software Engineering Design **Date**: 25 February 2004
Seminar Title: Performance Engineering of Real-time and Embedded Systems Seminar
Credit Hours: 4
Prerequisite(s): 4010-461
Co-requisite(s): None
Course proposed by: Jim Vallino

2.0 **Course information:**

	Contact hours	Maximum students/section
Classroom		
Lab		
Studio	4	24 (split with Comp. Engr.)
Other (specify _____)		

Quarter(s) offered *(check)*
Fall **Winter** ☒ **Spring** **Summer**

Students required to take this course: *(by program and year, as appropriate)*
None.

Students who might elect to take the course:
Upper division software engineering, computer engineering and computer science.

3.0 **Goals of the course** *(including rationale for the course, when appropriate)*:
This course has students explore performance aspects of real-time and embedded systems. Software engineering and computer engineering jointly developed this course as the second course in an embedded and real-time systems sequence. A NSF CCLI grant supports this work. This is the second offering of this course as a seminar.

4.0 **Course description** *(as it will appear in the RIT Catalog, including pre- and co-requisites, quarters offered)*
This course discusses issues of performance in real-time and embedded systems. Techniques for profiling the resource usage of a system and for measuring the effect of increasing system requirements will be covered. The control of physical

systems will motivate the need for performance tuning of a real-time system. Students will write programs running under a real-time operating system that can maintain control of a physical system. The course will discuss and experiment with performance trade-offs that can be made using hardware-software co-design. (4010-461) Class 4, Credit 4

5.0 Possible resources (texts, references, computer packages, etc.)

1. VxWorks operating system
2. Quanser experimental systems

6.0 Topics (outline):

- Performance measurements for real-time and embedded systems
- Profiling of program execution in embedded systems
- Exploration of linear control systems
- Interpretation of linear control parameters
- Hardware system description languages
- Hardware/software co-design

7.0 Intended learning outcomes and associated assessment methods of those outcomes

After taking this course a student will be able to:

1. Profile the execution of an embedded system
2. Distinguish differences between PID control modes.
3. Contrast effects of system parameters on control of a physical system.
4. Analyze the profiling data to determine which areas of the program would benefit most from performance tuning.
5. Compare performance of systems based on performance data.
6. Design a test and measurement plan to collect system performance data.
7. Describe hardware/software tradeoffs in the design of an embedded system.
8. Demonstrate the effects of moving the hardware/software boundary in a design

8.0 Program or general education goals supported by this course

This elective course supports the following software engineering program outcomes:

1. Apply mathematics, especially discrete mathematics, to the modeling and analysis of proposed and existing software systems.
2. Apply contemporary concepts (e.g., design patterns, frameworks, and architectural styles) to the design of moderately complex software systems (approx. 10,000 LOC).
3. Analyze proposed designs in terms of fundamental design principles (e.g., cohesion, coupling, information hiding, abstraction, and encapsulation).
4. Work in small teams (< 10 persons) to develop a software system.
5. Understand the ethical and professional responsibility incumbent upon them in different software development organizations (e.g., commercial off-the-shelf vs. safety-critical systems).

6. Write concise and correct technical documents relating to all phases of the software development cycle.
7. Make oral presentations of technical material.
8. Rapidly learn, assess, and adapt to new languages, environments, and paradigms for software development.
9. Relate principles of software engineering to at least one domain where those principles are or can be applied.

9.0 Other relevant information *(such as special classroom, studio, or lab needs, special scheduling, media requirements, etc.)*

Faculty from both software engineering and computer engineering will teach this course in the Real-Time and Embedded Systems Lab (70-1640) in the software engineering area.

10.0 Supplemental information

None.