SE 5303   Design Flows for Embedded/Networked Systems

Course Coordinator:  J. Chandy

Lecturers:    J. Chandy, P. Duggirala, S. Han

Course Objectives:
This course is designed to provide students with a thorough understanding of the design, verification, and validation of embedded/network systems and software-intensive systems. The student will develop skills in specifying requirements for embedded software systems, model based architecture and design, and verification and validation of embedded systems. Special emphasis will be placed on distributed embedded systems and real-time systems. Examples are driven by thermal fluid and electro-mechanical systems. The course is addressed to students in engineering who are pursuing the Embedded Systems track. The platform-based design (PBD) flow will be used as the common thread through the course.

Anticipated Student Outcomes
By the end of SE-5303, students will have achieved the following objectives:

1) Understand the platform-based design flow and how to apply it to practical embedded systems design from requirements, to architecture selection, to implementation, to verification.

2) Acquire ability to capture system-level requirements, modeling of requirements to support model-based development process, identification of missing or ambiguous requirements, and functional hazard analysis in the design of embedded systems;

3) Learn to evaluate, select and validate functional architecture, including system-level hazard analysis, allocation of functional and safety/reliability requirements to subsystems, states and transitions, and functional architecture analysis and evaluation methods;

4) Understand the issues involved with building real-time embedded systems including relevant requirements, system architectures, timing analysis, scheduling protocols, etc.

5) Understand the issues involved with building distributed embedded systems including relevant requirements, system architectures, distributed protocols, etc.

6) Command skills to conduct detailed embedded system design, including refined FMEA and reliability analysis, robustness verification, and test plan;

7) Build skills to perform embedded system verification and validation, including MIL, SIL, and HIL-based testing, verification, validation; and performance and robustness verification;

8) The course is intended to serve as a key component to achieve standard work proficiency levels L2-L3 in embedded system design.
Pre-requisites

- SE 5301
- SE 5302

Course Organization

The course is organized into three learning modules:

**Learning Module I (Lectures 1-4): Embedded Systems Architecture and System Design**

This module is meant to give students an introduction to system design and its interaction with embedded systems design. It is composed of four lectures devoted to understanding system level requirements, functional architecture selection, and control structure evaluation. The module will elaborate on embedded systems models and formal methods that were introduced in earlier courses. Embedded systems se cases will be introduced and will be used throughout the course to exemplify the types of challenges faced during product development.

**Learning Module II (Lectures 5-11): Detailed Embedded Systems Design**

This module is meant to give students an understanding of embedded systems design within the context of a large system – particularly the implications of real-time systems and distributed systems. Students will develop skills to conduct preliminary system design, including FMEA and system level safety analysis, sensitivity analysis, reliability analysis and security assessments.

**Learning Module III (Lectures 12-14): Embedded Systems Verification and Validation**

This module is devoted to understanding how to verify and validate an embedded system design. The four lectures in this module develop skills to perform embedded system verification and validation, including MIL, SIL, and HIL-based testing, simulation, and formal verification.

Course Outline

**Lecture 1: Introduction to Product Development Processes for Embedded Systems**

- Embedded Systems – Control Systems, Chiller Systems in buildings, ECS for aircraft
  - High-level description of functions, principles of operation, components, design options and performance metrics
• Current industry product development process
• Platform-based design principles for embedded systems design
• Focus of this Course: Requirements, Robustness, Safety, Requirements Compliance

Lecture 2-3: System Level Requirements Capture for Embedded Systems

• Decomposition and allocation of system requirements
  o Standard work at BIS using functional decomposition
• Types of requirements: functionality, interfaces, performance, attributes, communication, and constraints (memory, power, cost, etc.)
• How to capture requirements
  o Natural language (current state)
  o Models
• Value of requirements analysis and relationship to formal methods
• Review of basic models
  o FSM
  o System functional models
  o Constraint models
  o Functional block diagrams – SysML, state machines (HW and SW)
• Which type of models to use for different types of requirements
• How do you relate different models and how do they affect design process?
• Requirement templates to guide identification of missing or ambiguous requirements
• SysML/UML/SystemC/Simulink Modeling of requirements to support model-based development process
• Capture real-time requirements (MARTE,AADL)
• Identification of subsystem and functional requirements
• Application to ECS control system

Team Task (HW):
Develop SysML models of embedded systems requirements for chiller control systems

Recommended Readings:
• Models of computation: Lee & Varaiya Structure and Interpretation of Signals and Systems - (open access http://leevaraiya.org/) – more specifics on which chapters
• SysML/UML: UML distilled - Martin Fowler (http://martinfowler.com/books/uml.html)
• Marte: Specification and tutorials from http://www.omgmar-te.org/

Lecture 4: Architecture Evaluation, Selection and Validation

• PBD approach of architectures
• Function/control
• Embedded
• Communication
• Architecture views – performance, safety, reliability, etc.
• Concept of “verify early” – little “V”
• Platform-based component architectures
• Use cases
• Hazards and reliability allocation to subsystems
• Hardware-software partitioning and codesign
• Computation Independent Model (CIM)/Platform Independent Model (PIM)
• Metrics for evaluation
• Methods to evaluate functional architectures
  • Executable SysML/UML models
  • SystemC simulations
  • Quality Function Development (QFD)
• Synthesis and Design Space Optimization
• Architecture Instance Mapping
• Reinforce use of languages – SysML, UML
• Application to ECS control system

Team Task (HW):
Apply functional architecture selection, evaluation, and validation process to chiller control systems

Recommended Readings:
• Sangiovanni-Vincentelli, A., 2007, Quo vadis, SLD? reasoning about the trends and challenges of system level design, Proceedings of the IEEE 95 (3), 467-506
• Other PBD papers

Lecture 5: Design for Safety, Reliability, Availability, and Security

• Safety risk assessment
• FMEAs, Fault Trees, and functional hazard analysis (FHA)
• Fault tolerance and redundancy
• Safety certification standards
  • Aerospace: DO-178C, ARP4754, etc.
  • Elevators and fire safety – EN81, UL. Also, IEC61508 safety software standard which is typically used in elevator systems.
Lecture 6: Design for Security

- Security risk assessment
- Secure embedded system design
  - Secure software engineering
  - Access control
  - Hardware security
  - Security countermeasures

Lecture 7-9: Distributed and Networked Embedded Systems Design

- Network properties
  - Network access
  - Supported topologies
  - Bandwidth
  - Reliability
  - Real-time performance – latency, jitter, coherence
  - Security
- Network architectures
  - OSI framework – descriptions of layers
  - Physical layer options including wireless
    - Copper, coax, fiber
    - 802.3 for Ethernet
    - 802.11 for WiFi
    - 802.15.4 for ZigBee, WirelessHART and ISA 100.11a
    - Cellular and satellite
  - Embedded system communication protocols
    - Types of protocols – MAC (CSMA, CDMA, TDMA, priority, etc.)
    - Protocol examples
      - Ethernet
        - Physical topology
        - Ethernet frame structure
        - Ethernet’s MAC protocol
        - Ethernet switch
      - CAN
- Basic concepts
- Bus access and arbitration
- CAN frame formats
- Error detection and handling
- CAN controller

**TTP**
- Time-triggered vs. event-triggered protocol
- TTP system structure
- TTP/A and TTP/C protocol layers
- TTP time synchronization
- Frame formats
- TT/Ethernet vs. TTP vs. Ethernet

**BACnet**
- Automated Logic WebCTRL
- Carrier control
- BACnet/IP
- TCP/IP stack (Some topics are optional, depending on time availability)
  - Transport layer services
  - Multiplexing and demultiplexing
  - Connectionless transport - UDP
  - Principle of reliable data transfer
  - Connection-oriented transport – TCP
  - Principle of congestion control
  - TCP congestion control
  - Virtual circuit and datagram networks
  - Router inside
  - IP: Internet protocol
  - Routing algorithms
  - Routing in the Internet
  - Broadcast and multicast routing

**Network services** (Some topics are optional, depending on time availability. Focusing on network services for networked embedded systems)
- Web
- Web services – SOAP, JSON, etc.
- Deterministic TCP/IP
- FTP
- Electronic mail
- DNS
- P2P applications
- Socket programming with UDP and TCP
- Database connection
- Modeling and analysis for embedded system communication protocols
- Model-based evaluation of distributed protocols – TEMPO? (Note: I am not familiar with this topic and need some time to look into details.)
- Construct communications model for networked ECS control system

**Team Task (HW)**
- communications model for chiller control systems

**Suggested Readings:**
  - Chapter 2 for application layer including Web/HTTP, FTP, Email, DNS and P2P protocols;
  - Chapter 3 for transport layer including TCP and UDP;
  - Chapter 4 for IP;
  - Chapter 5 for Ethernet;
  - Chapter 6 for 802.11. The reference for database depends on which commercial or open-source database we are going to use for the web application.
- CAN bus website: http://www.canbus.us/
- IEEE 802.3 Ethernet working group: http://www.ieee802.org/3/
- IEEE 802.11 wireless local area networks: http://www.ieee802.org/11/
- IEEE 802.15.4 WPAN: http://standards.ieee.org/findstds/standard/802.15.4-2006.html
- IEEE 802.15 WPAN Task Group 4e: http://www.ieee802.org/15/pub/TG4e.html
- ISA-100 Wireless Compliance Institute: http://www.isa100wci.org/

**Lecture 10-11: Real-Time Systems Design**
- Software Architectures for Real-Time Systems
- Timing requirements, and timing analysis
  - Concept of worst-case Execution Time (WCET)
    - Variance in execution time
• Probabilistic timing analysis
  o Why it is hard for analyzing WCET?
  o Overall Approach: Modularization
  o Program Path Analysis
  o Static Analysis
• Uniprocessor scheduling algorithms
  o Task models
  o Performance metrics of scheduling algorithms
  o Static scheduling algorithms
  o Dynamic scheduling algorithms
    ▪ Static-priority scheduling
    ▪ Dynamic-priority scheduling
• Multicore and distributed systems issues in RT
  o Multiprocessor scheduling
    ▪ Partitioned scheduling
    ▪ Global scheduling
    ▪ Semi-partitioned scheduling
  o Multicore resource sharing
  o Distributed real-time systems
• What happens when things go wrong – e.g. when deadlines are missed, when system not schedulable? How do degrade gracefully?
• Application to ECS

Team Task (HW):
• Apply detailed design process to chiller

Suggested Readings:

Lecture 12-13: Overview of Verification & Validation

• Validation vs Verification
• Requirements validation versus system validation
- Requirements decomposition and allocation
- Requirements and Verification impact
- V & V Completeness criteria

• Validation and Verification Methods: Models, review, xIL, analysis, testing
  - Virtual prototyping
  - Model types (behavioral, steady state, dynamic, etc.)
  - Analysis methods and role in verification: schedulability, worst case analysis
  - xIL usage for systems, software and hardware
  - Engineering review processes
  - Verification by similarity, field history and other methods
  - Test based verification

• Traditional V model versus PBD design flow
  - History of V&V and application to BIS, UTAS
  - Review of PBD Design flow – progression from model to hardware
  - When to use simulation vs. xIL verification vs. formal verification?

• Review of Formal Verification
  - Static program verification
  - Application to chiller control system
  - Model checking and timing verification
  - Formal verification
    - Case study with tools
    - Relationship to standards

• Additional Model Based verification methods
  - Formal methods
  - Automated Test generation methods
  - Model based coverage analysis
  - Robustness analysis and examples

• How the different techniques fit together?
  - Practical selection of appropriate method of verification
  - Cost impact

Team Task (HW):
- Apply component and system level validation process to ECS control systems

Recommended Reading:
- Principles of Cyber-Physical Systems, Rajeev Alur, MIT Press, 2015,
  http://mitpress.mit.edu/books/principles-cyber-physical-systems, Chapter 5
Lecture 14: System Testing

- MIL, SIL and HIL-based verification
- Component testing
- Real-time systems verification
- Distributed systems verification
- Automated model-based test generation
USEFUL READING

Primary course textbook


Software References (available free online)


Other Reference Books (available free online)


Other Reference Books (for purchase)


Helpful links:

- Virtual Computer Lab at UConn: [http://skybox.uconn.edu/](http://skybox.uconn.edu/)
- Course Material: [https://lms.uconn.edu](https://lms.uconn.edu)
- Institute for Advanced Systems Engineering: [http://www.utc-iase.uconn.edu/](http://www.utc-iase.uconn.edu/)
Coursework Targeting Student Outcomes

During the semester, students will be challenged in areas that are designed to help them to successfully realize proficiency in the student outcomes: Participation, Homework, Oral Presentations, and Project Report. The final course grade will be based on the following:

Grading

- Homework assignments, 70%
  - We expect approximately one assignment per week over the duration of the course.
- Final project/exam, 30%
  - The final project / final exam will be similar to the homeworks, but extended and incorporating material from the three modules.

Homework

Homework assignments will be posted on HuskyCT. Homework assignment due dates will be given with the assignment. NO late homework will be accepted as the homework will often be discussed in class. Each problem will be graded on a scale of 0-100.

Project, Presentations and Project Report

A project is to be developed by student groups, which is expected to evolve during the entirety of the track. The portion of the project that is to be executed in this course refers mainly to design project identification, challenge quantification, significance and relevance to the MBD philosophy and plan of attack. The final deliverable (presentation) should identify all the aforementioned elements in a quantifiable manner and suggest a strategy for solution.
Other Policies

Student Conduct:

http://www.dosa.uconn.edu/student_code.html. Students are responsible for adherence to the University of Connecticut student code of conduct. Perhaps the most important policy to pay attention to is the section on Student Academic Misconduct. “Academic misconduct is dishonest or unethical academic behavior that includes, but is not limited, to misrepresenting mastery in an academic area (e.g., cheating), intentionally or knowingly failing to properly credit information, research or ideas to their rightful originators or representing such information, research or ideas as your own (e.g., plagiarism).” Examples of academic misconduct in this class include, but are not limited to: copying solutions from the solutions manual, using solutions from students who have taken this course in previous years, copying your friends’ homework, looking at another student’s paper during an exam, lying to the professor or TA and incorrectly filling out the student workbook.

Attendance:

Attendance will not be taken; however, it is practically impossible to follow the class if classes are missed.

Absences:

Make-up of missed exams requires permission from the Dean of Students, see “Academic Regulations.” Midterm-exams are treated the same as Final Examinations. Students involved in official University activities that conflict with class time must inform the instructor in writing prior to the anticipated absence and take the initiative to make up missed work in a timely fashion. In addition, students who will miss class for a religious observance must “inform their instructor in writing within the first three weeks of the semester, and prior to the anticipated absence, and should take the initiative to work out with the instructor a schedule for making up missed work.”

Course Schedule*

<table>
<thead>
<tr>
<th>Date</th>
<th>Lecture</th>
<th>References</th>
<th>Lecturers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Introduction to Product Development Processes for Embedded Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>System Level Requirements Capture for Embedded Systems – Part I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Course Title</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Level Requirements Capture for Embedded Systems – Part II</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architecture Evaluation, Selection and Validation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design for Safety, Reliability, Availability, and Security</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design for Security</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributed and Networked Embedded Systems Design – Part I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributed and Networked Embedded Systems Design – Part II</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributed and Networked Embedded Systems Design – Part III</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Real-Time Systems Design – Part I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Real-Time Systems Design – Part II</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overview of Verification &amp; Validation – Part I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overview of Verification &amp; Validation – Part II</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Testing</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>