SWE 760
Lecture 2:

Structural Modeling for RT Embedded Systems

Reference:
H. Gomaa, Chapters 4, 5 - *Real-Time Software Design for Embedded Systems*, Cambridge University Press, 2016

Hassan Gomaa
Dept of Computer Science
George Mason University
Fairfax, VA

GEORGE MASON UNIVERSITY

Introduction

- Integrate
  - Systems Modeling
  - Software Modeling
- Development of RT embedded systems
- Approach from system and software modeling perspective
- COMET/RTE
  - Extension of COMET method
  - Real-Time Software Modeling and Design with UML
  - From Use Case Models to RT Software Architecture
- *H. Gomaa, Real-Time Software Design for Embedded Systems*, Cambridge University Press, 2016

Static Modeling

- Define structural relationships between classes
  - Depict classes and their relationships on class diagrams
- Relationships between classes
  - Associations
  - Composition / Aggregation
  - Generalization / Specialization
- Static Structural Modeling for Real-Time Embedded Systems
  - Conceptual Structural Model of System
    - Model hardware, software, people, systems
  - System Context Diagram
    - Depict total (hardware/software) system boundary
  - Software System Context Diagram
    - Depict software system boundary

Figure 2.3: UML notation for relationship on class diagram

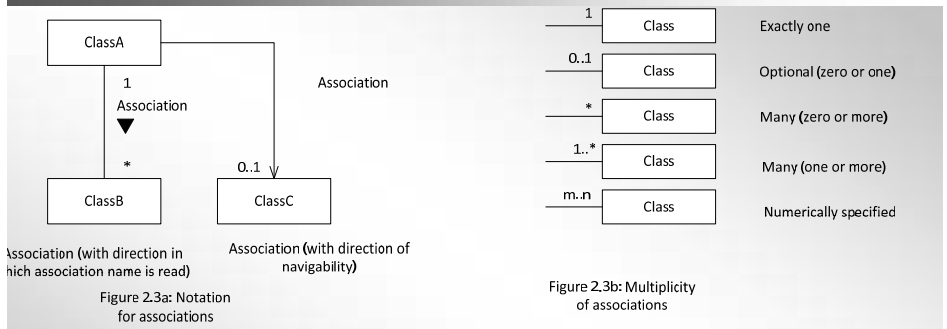


Figure 2.3a: Notation for associations

Figure 2.3b: Multiplicity of associations

Figure 2.3c: Generalization/ specialization

Figure 2.3d: Aggregation hierarchy

Figure 2.3e: Composition hierarchy

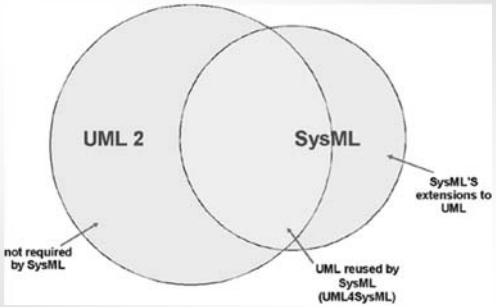
Modeling Approach

- Modeling languages
  - SysML for system (hardware/software) modeling
  - UML for software modeling
- SysML
  - Subset of UML 2 with extensions for system engineering
  - Use SysML subset compatible with UML 2
    - Block definition diagrams (static modeling)
    - State machine diagrams
    - Use case diagrams
- Examples of simple embedded system
  - Microwave Oven System
  - Railroad Crossing System

SysML™

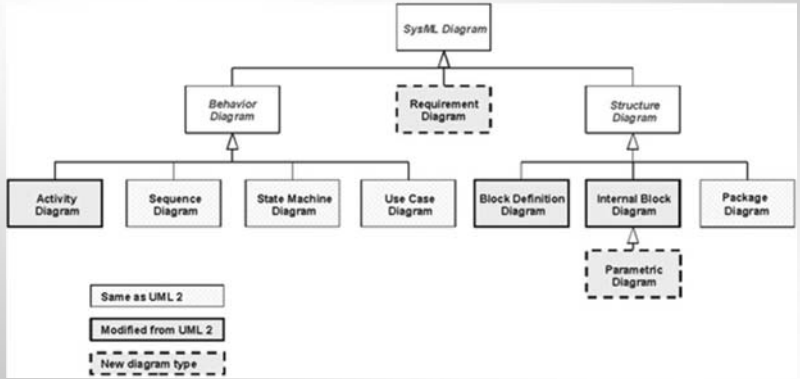
- Systems Modeling Language
  - Standardized notation for modeling system requirements
  - Approved as a standard by Object Management Group (OMG)
  - Methodology independent
- General-purpose graphical modeling language
  - specifying, analyzing, designing, and verifying complex systems
    - Hardware
    - software
    - information
    - personnel,
    - Procedures, facilities.
  - Provides graphical representations for modeling
    - system requirements
    - Behavior
    - Structure
    - parametrics

Relationship between SysML and UML
Source: OMG



- SysML
  - Subset of UML 2
  - Extensions to UML for systems modeling

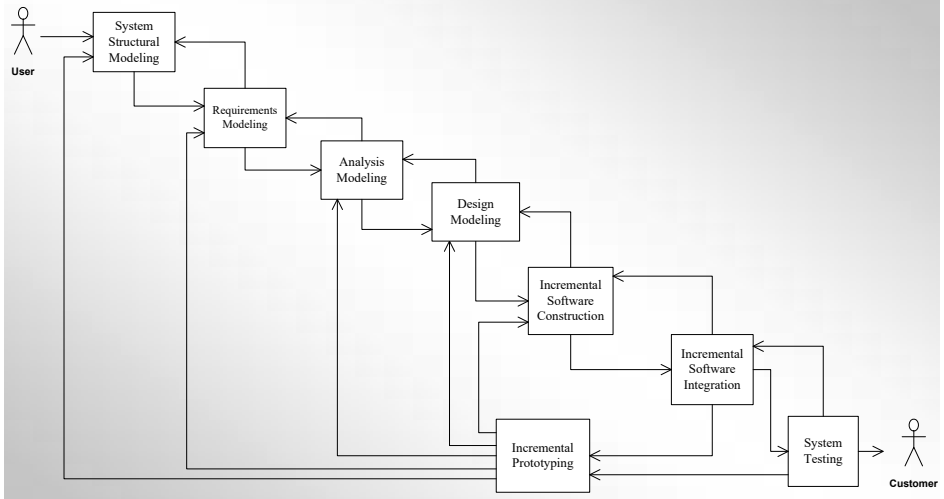
SysML Diagram Types
Source: OMG



Subset of SysML for Systems Analysis

- Block Definition Diagram
  - Based on UML class diagram
  - Use for structural modeling of system
- Same as UML
  - Use Case diagram
  - State Machine diagram
  - Sequence diagram
- Requirements diagram could be used
  - More concise to use requirements tables

Figure 4.1 COMET/RTE life cycle model



System Structural Modeling

1. Structural Modeling of the Problem Domain
2. Structural Modeling of the System (hardware/software) Context
3. Hardware/Software Boundary Modeling
4. Software Context Modeling
5. System Deployment Modeling

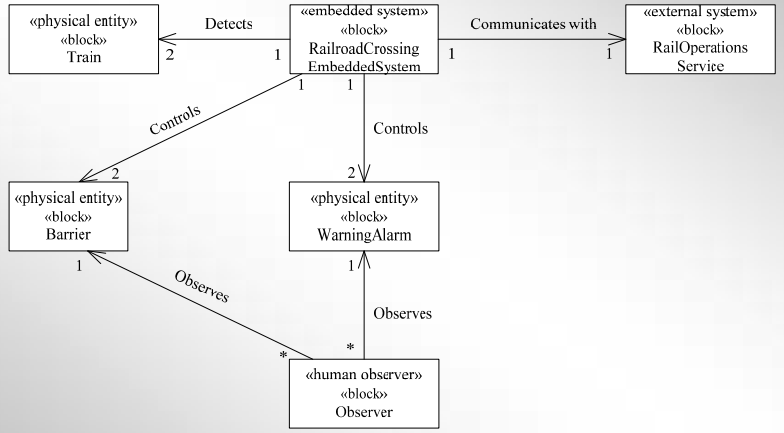
Structural Modeling of Problem Domain

- Model real-world entities relevant to embedded system
  - External to total hardware/software system
- Develop conceptual static model for system
- Use SysML block definition diagram
  - Block is more general concept than class
    - Class is software concept
    - Block could be hardware or software entity
  - Same notation as class diagram
    - Exploit power of static modeling
  - Blocks are stereotyped classes
  - Block relationships same as class relationships

Structural Modeling of Problem Domain

- System under development
  - Model as «embedded system» block
- Physical entity
  - Has physical characteristics (can see, touch)
  - Model as «physical entity» block
  - Detected and/or controlled by system
- Existing system
  - Model as «external system» block
  - Communicates to and/or from Embedded System
- User - interacts with the system
  - Model as «human user» block
- Observer - observes but does not use the system
  - Model as «human observer» block

Fig. 5.6: Conceptual Structural Model of Railroad Crossing Problem Domain



Block Definition Diagram

Structural Modeling of System Context

- Defines boundary between total system (hardware and software) and external environment
  - Depicted on System Context Diagram
    - SysML block definition diagram
    - Uses UML static modeling notation
- System
  - Depicted as one aggregate «system» block
- External environment
  - External blocks that system interfaces to

System Context Modeling

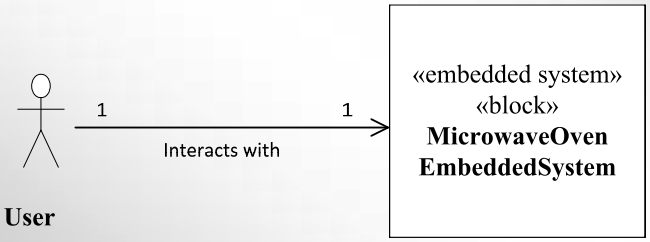
- Categories of external blocks
- Depicted using UML stereotypes
  - «external physical entity»
    - Detected or controlled by system
    - Does not physically connect to system
      - Needs sensors or actuators to make physical connection
  - «external system»
    - Communicates to and/or from System
  - «external user»
    - Uses the system
  - «external observer»
    - Observes but does not use the system

Associations on System Context Diagram

- System Context Diagram shows
  - Association between system and external block
  - Multiplicity of association (1 to 1, 1..* to 1, etc.)
- Associations have standard names
  - «embedded system» *Outputs to* «external user»
  - «external physical entity» Inputs to «embedded system
  - «embedded system» *Detects* «external physical entity»
  - «embedded system» *Controls* «external physical entity»
  - «external observer» *Observes* «embedded system»
  - «external user» *Interacts with* «embedded system»
  - «external system» *Communicates with* «embedded system»

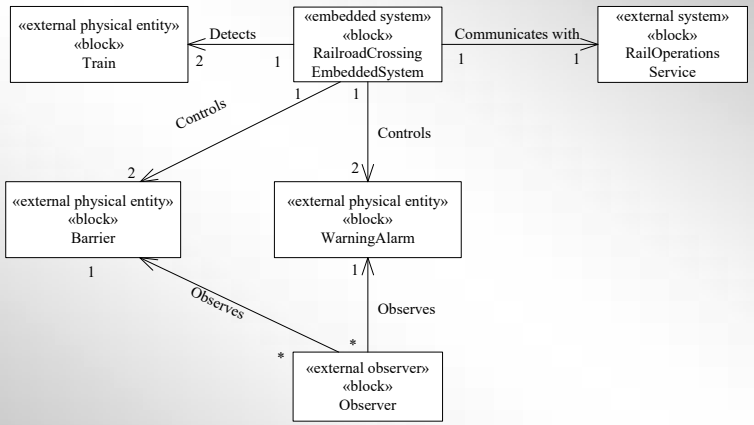
System Context Diagram for Microwave Oven System

Figure 19.2: Microwave Oven System context diagram



SysML Block Definition Diagram

Fig. 5.7: System Context Diagram for Railroad Crossing Embedded System



Block Definition Diagram

Hardware/Software Boundary Modeling

- Embedded system
  - Part of a larger hardware/software system
  - Interacts with external environment through sensors and actuators
  - Sensor
    - Input device that senses changes to external body
    - E.g., Door sensor senses
      - door opened, door closed
  - Actuator
    - Output device that effects changes to external body
    - E.g., Actuator switches Heating Device on and off
- Model composition of embedded system

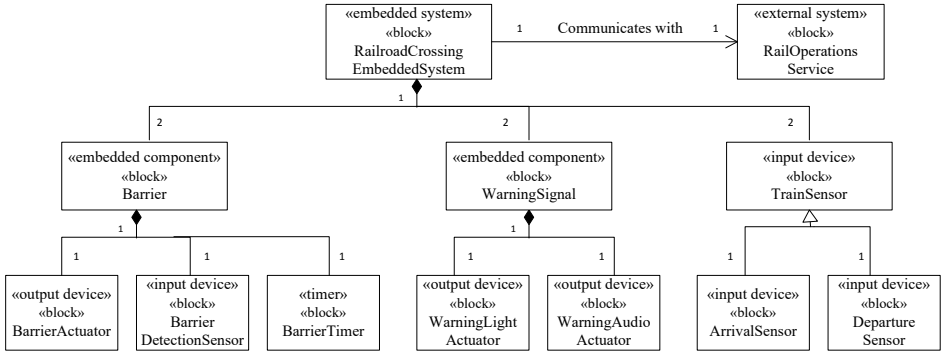
Hardware/Software Boundary Modeling

- Decomposition into hardware and software components
  - Physical hardware components are explicitly modeled on the hardware/software block diagram
  - Software system is modeled as one composite component
  - Physical entities of the problem domain are often input and/or output hardware devices that interface to the software system
- External physical entity
  - Does not physically connect to system
  - Needs sensors or actuators to make physical connection

Example of Hardware/Software Boundary Modeling

- External physical entities – Train, Barrier, Warning Alarm
- Arrival and Departure of Train need sensor detection
  - Arrival Sensor, Departure Sensor
- Control of Barrier needs
  - Barrier Actuator to raise and lower barrier
  - Barrier Detection Sensor
    - To detect when barrier completes raising & lowering
- Control of Warning Alarm needs
  - Warning Light Actuator
  - Warning Audio Actuator

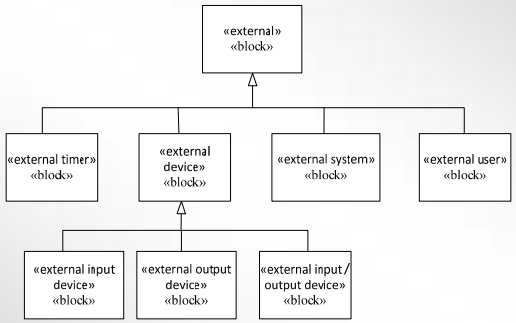
Figure 20.2 Structural model for Railroad Crossing Control System



Software System Context Modeling

- Defines boundary between software system and external environment
  - Depicted on Software System Context Diagram
- Software System
  - Depicted as aggregate «software system» block/class
- Categories of external blocks that system interfaces to
  - Depicted using UML stereotypes
    - «external I/O device»
    - «external user»
    - «external system»
    - «external timer»

Classification of external blocks by stereotype (Fig. 5.8)



Associations on Software System Context Diagram

- Software System Context Diagram shows
  - Association between software system and external block
  - Multiplicity of association (1 to 1, 1..* to 1, etc.)
- Associations have standard names
  - «external input device» *Inputs to* «software system»
  - «software system» *Outputs to* «external output device»
  - «external user» *Interacts with* «software system»
  - «external system» *Communicates with* «software system»
  - «external timer» *Signals* «software system»

Fig. 5.9: Software System Context Diagram for Microwave Oven System

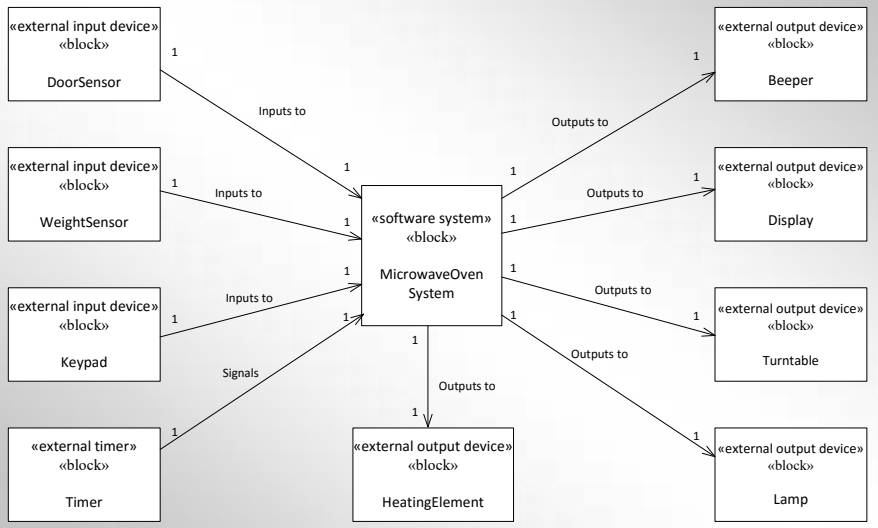
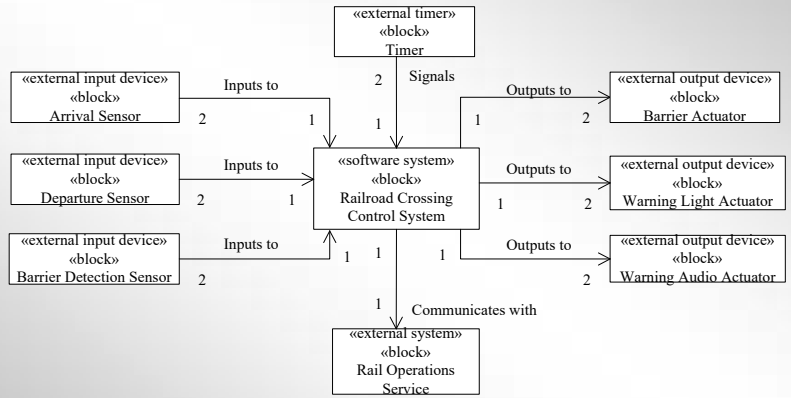


Fig. 5.10: Software System Context Diagram for Railroad Crossing Control System



Defining Hardware/Software Interfaces

- Define interface between hardware devices and software system
- Specification of hardware/software interface for each hardware device
  - **Name** of I/O device:
  - **Type** of I/O device:
  - **Function** of I/O device:
  - **Inputs** from device to software system:
  - **Outputs** from software system to device:

Example of I/O device boundary specification

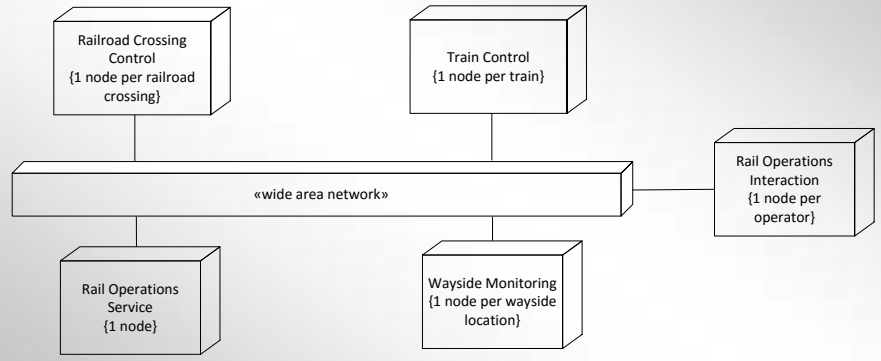
- Input device:
  - **Name** of I/O device: Arrival Sensor
  - **Type** of I/O device: Input
  - **Function** of I/O device: Signals when train arrives
  - **Inputs** from device:
    - Arrival Event
  - **Outputs** from software system to device: None

Defining Hardware/Software Interfaces Examples of I/O device specifications

- Output device
  - **Name** of I/O device: Barrier Actuator
  - **Type** of I/O device: output
  - **Function** of I/O device: Raises and lowers barrier
  - **Inputs** from device : None
  - **Outputs** to device:
    - Raise Barrier, Lower Barrier

System Deployment Modeling

- Physically deploy hardware and software components to
  - Hardware and software platforms
- Example from Distributed Light Rail Embedded System (Fig. 5.11)



Review of System Structural Modeling

1. Structural Modeling of the Problem Domain
2. Structural Modeling of the System (hardware/software) Context
3. Hardware/Software Boundary Modeling
4. Software Context Modeling
5. System Deployment Modeling

Figure 4.1 COMET/RTE life cycle model