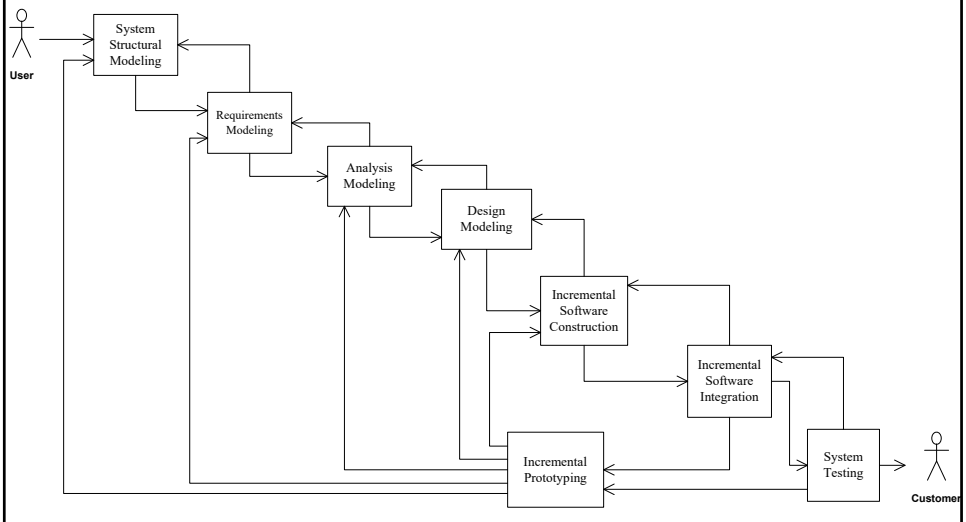
SWE 760
Real-Time Software Design

Lecture 6 – Dynamic Interaction Modeling for Real-Time Embedded Systems

Reference: H. Gomaa, Chapters 9 - *Real-Time Software Design for Embedded Systems*, Cambridge University Press, 2016

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

Figure 4.1 COMET/RTE life cycle model



Analysis Modeling

- Static Modeling
  - Define entity classes and relationships
- Dynamic State Machine Modeling
  - Real-time systems are highly state dependent
  - Actions depend on input event AND current sta**te**
- Object Structuring
  - Determine objects that realize each use case
- **Dynamic Interaction Modeling**
  - **Determine sequence of interactions among objects**

Dynamic Interaction Modeling

- Use cases realized in Dynamic Model
  - Show objects participating in each use case
- Determine how objects participate in use case
  - Use object structuring criteria to determine objects
    - Stereotype for each object structuring criterion
  - Shows sequence of object interactions in use case
    - Depict on
      - **communication diagram** or
      - **sequence diagram**
- State dependent control objects
  - Modeled using statecharts
- Dynamic Modeling
  - Approach to determine how objects interact with each other to support use case

Dynamic Modeling

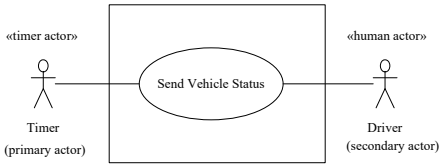
- Determine how objects interact with each other to support use case
  - Start with external event from actor
  - Determine objects needed to support use case
  - Determine sequence of internal events following external event
  - Depict on communication diagram or sequence diagram
- Stateless (non state-dependent) Dynamic Modeling
- State dependent Dynamic Modeling

Stateless Dynamic Modeling

- Start with use case
- Determine boundary objects
  - Receives external events from actor
- Determine internal objects
  - Receive messages from boundary objects
- Determine object interactions
  - Sequence of messages passed
- Develop main interaction sequence (scenario)
- Develop alternative sequences
  - For alternative branches of use case
  - E.g, for error handling or less frequently occurring conditions

Example of Stateless Dynamic Modeling

Figure 9.4 Send Vehicle Status use case



Example of Stateless Dynamic Modeling

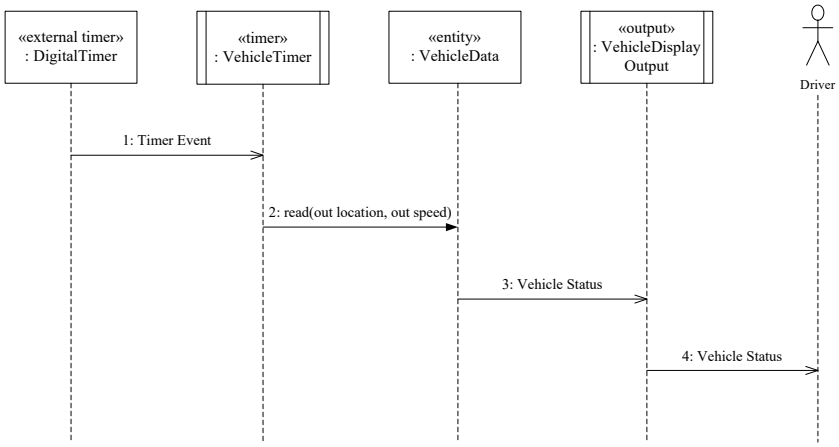
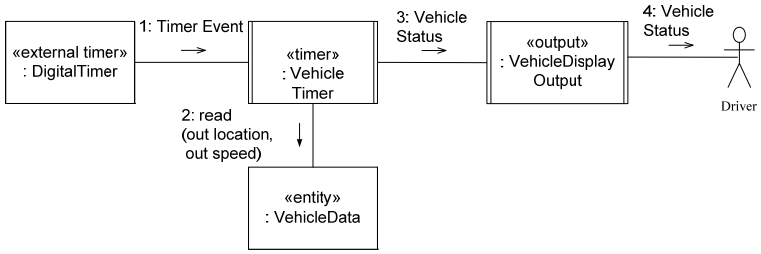


Fig 9.5: Sequence diagram for Send Vehicle Status use case

Example of Stateless Dynamic Modeling

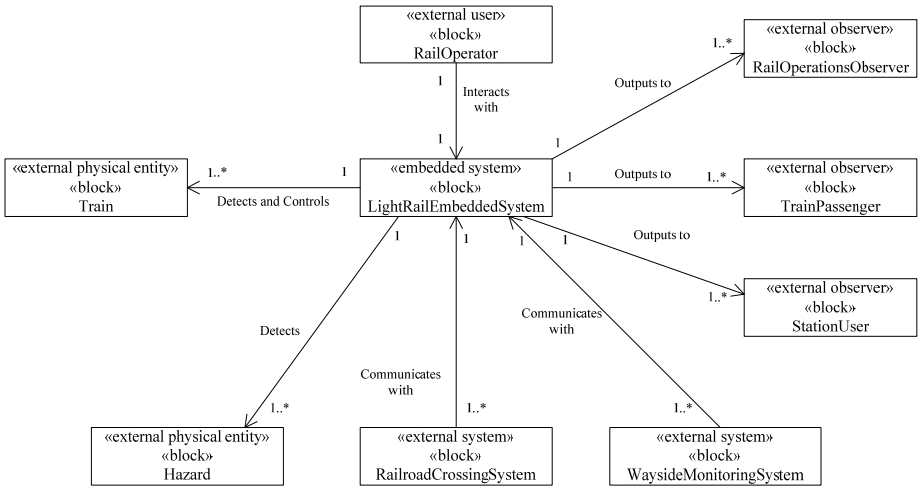
Figure 9.6: Communication diagram for Send Vehicle Status use case



State Dependent Dynamic Modeling

- Object interaction controlled by statechart(s)
  - Control object
    - Executes statechart
    - Activates/deactivates other objects
- For each use case
  - Determine objects participating in use case
  - Determine sequence of object communication
  - Develop statechart for control object
- For each event that arrives at control object
  - Determine state transition from current state to next state
  - Determine actions and activities to be executed
  - Determine objects required to perform actions and activities

Figure 21.2 Light Rail Control System context block diagram



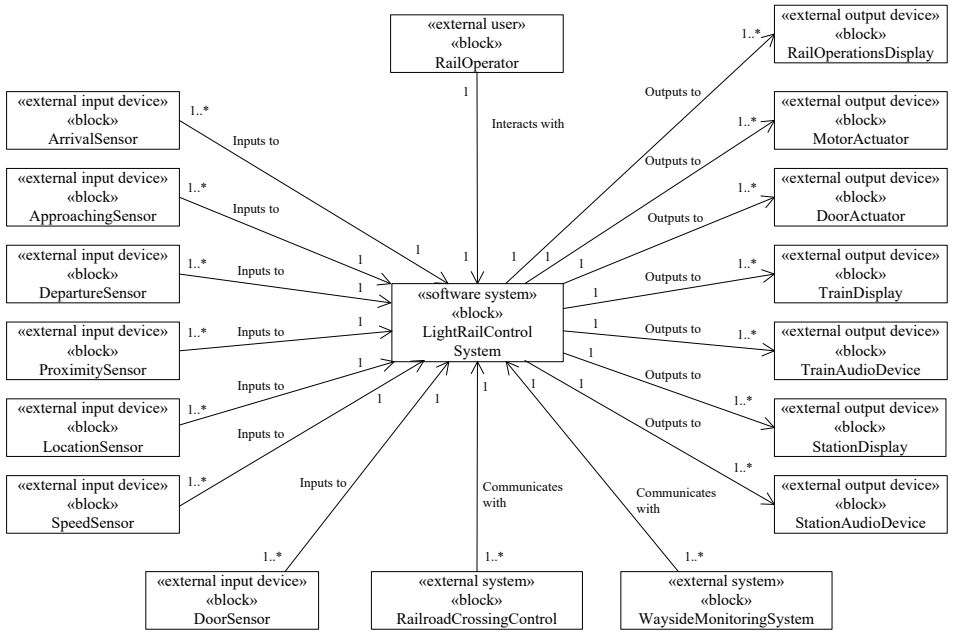
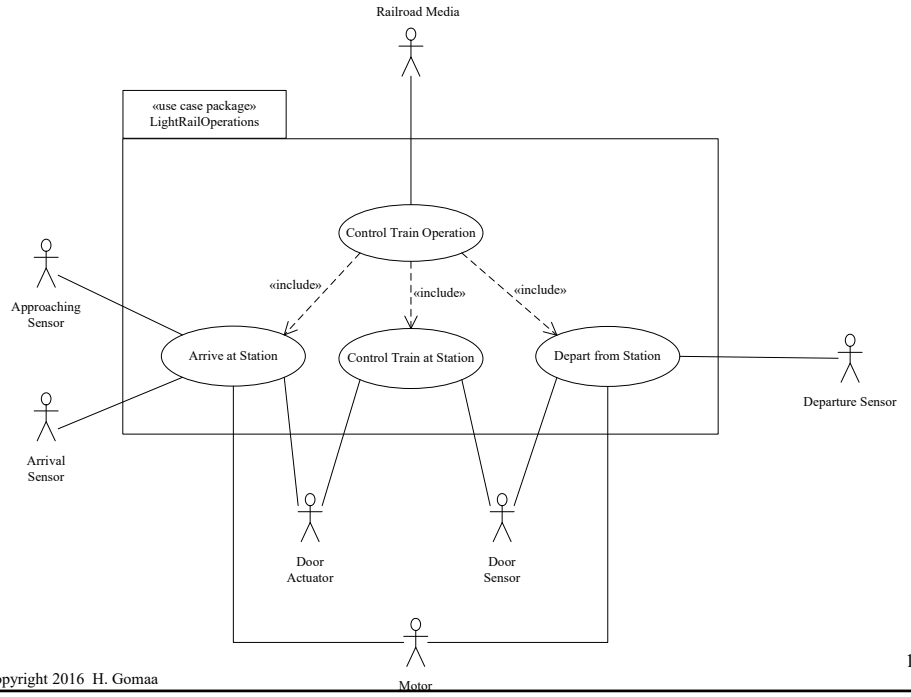


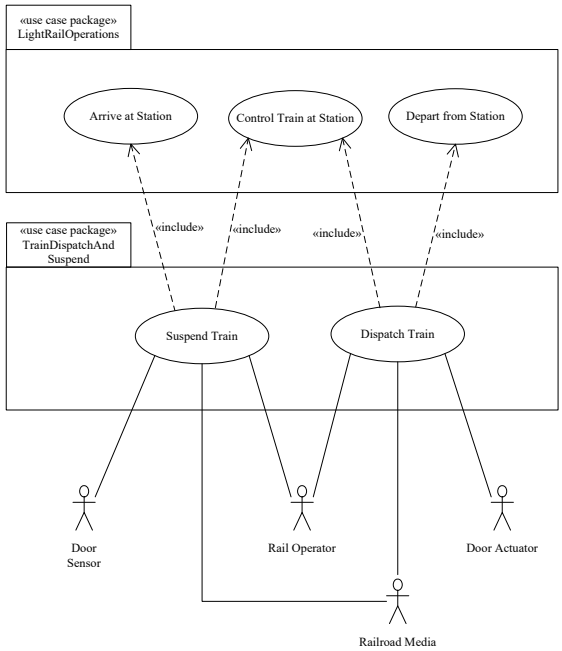
Figure 21.3 Light Rail Control Software system context block diagram

Figure 21.4 Light Rail Control System actors and use cases: Light Rail Operations use case package



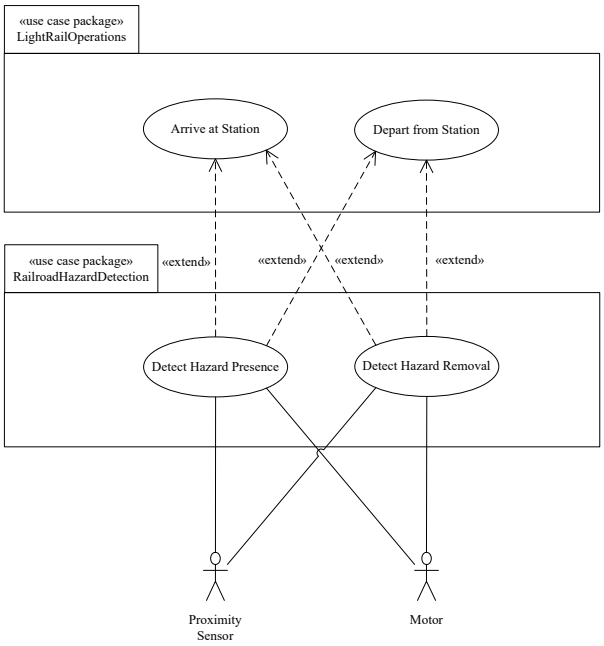
13

Figure 21.5 Light Rail Control System: Train Dispatch and Suspend use case package



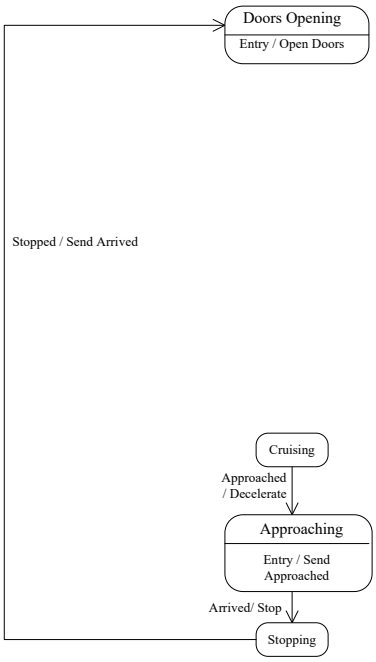
14

Figure 21.6 Light Rail Control System: Railroad Hazard Detection



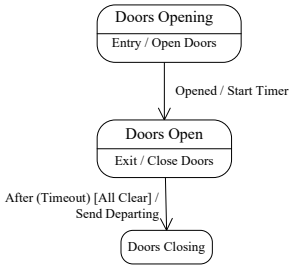
15

Figure 21.9 State machine for Arrive at Station use case



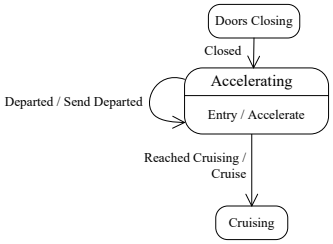
16

Figure 21.10 State machine for Control Train at Station use case



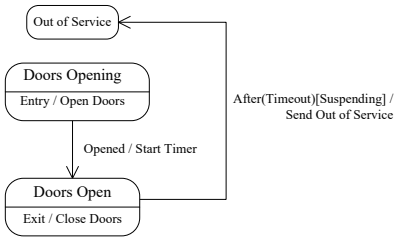
17

Figure 21.11 State machine for Depart from Station use case



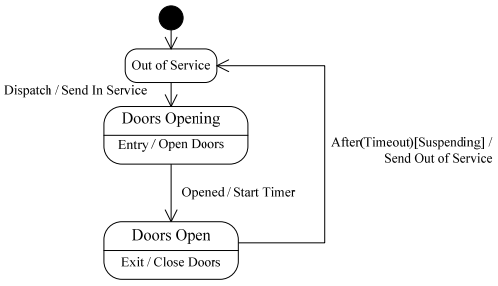
18

*Figure 21.12 State machine for **Suspend Train** use case*



19

*Figure 21.13 State machine for **Dispatch Train** use case*



20

Figure 21.14 State machine for Detect Obstacle Presence use case

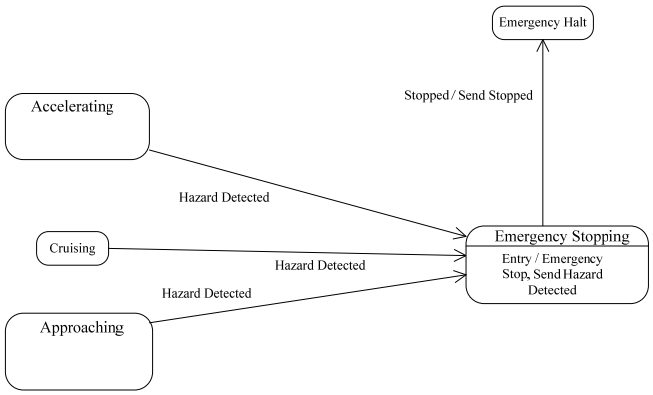
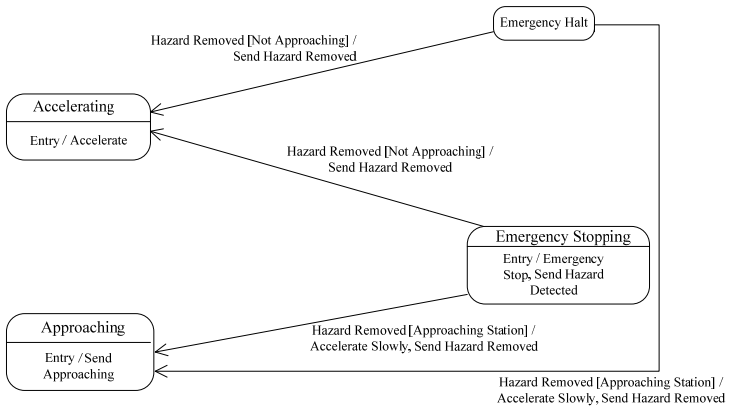
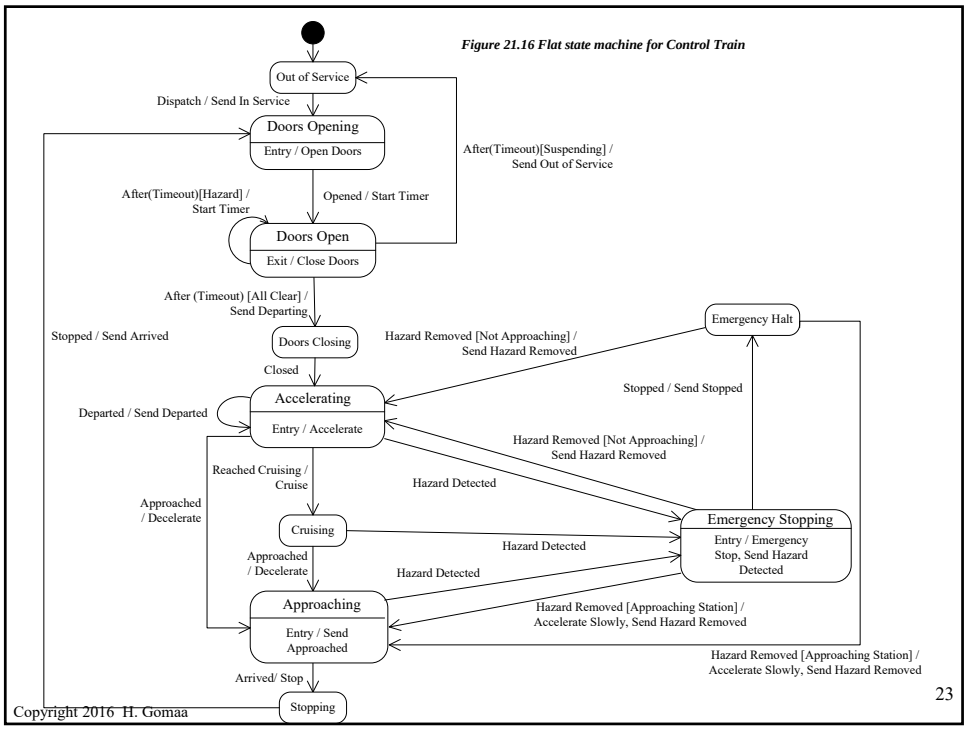
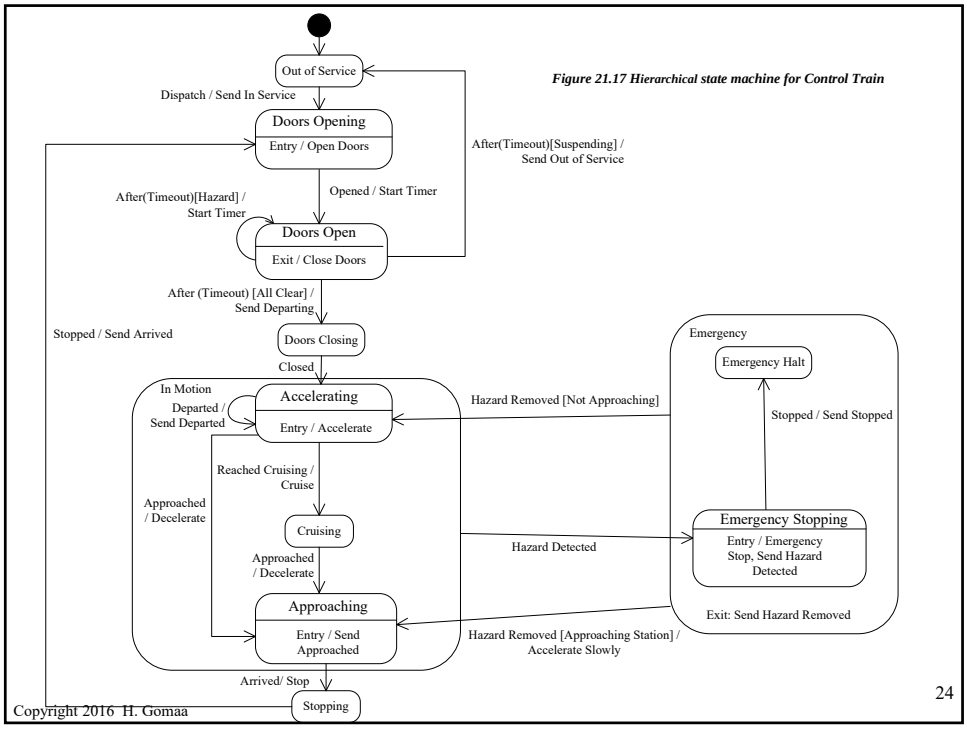


Figure 21.15 State machine for Detect Obstacle Removal use case





23



24

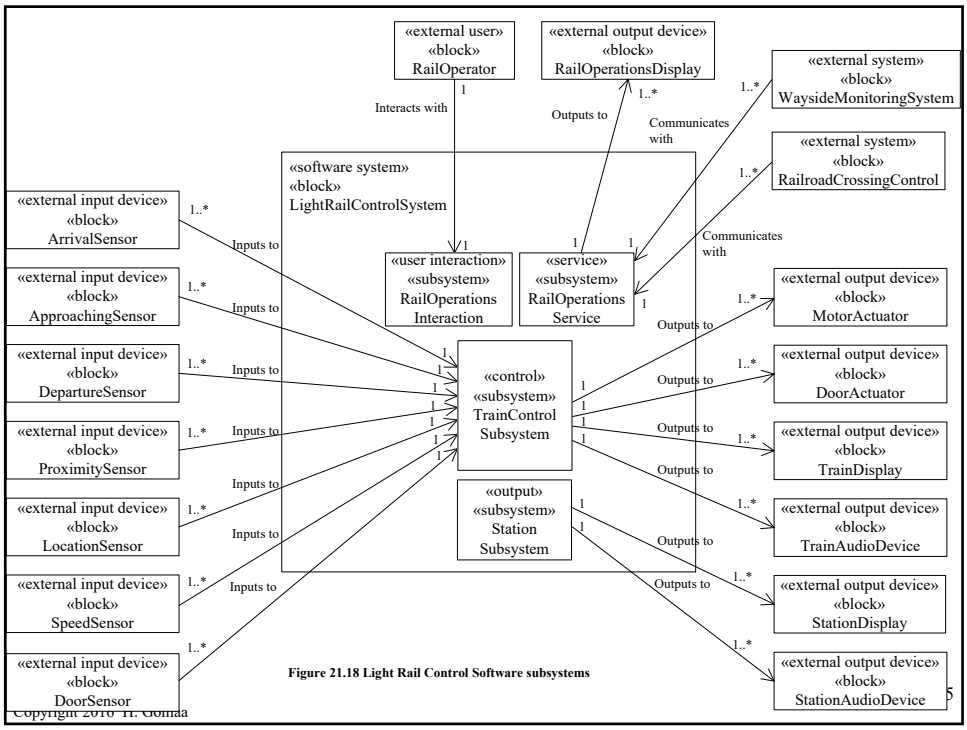


Figure 21.18 Light Rail Control Software subsystems

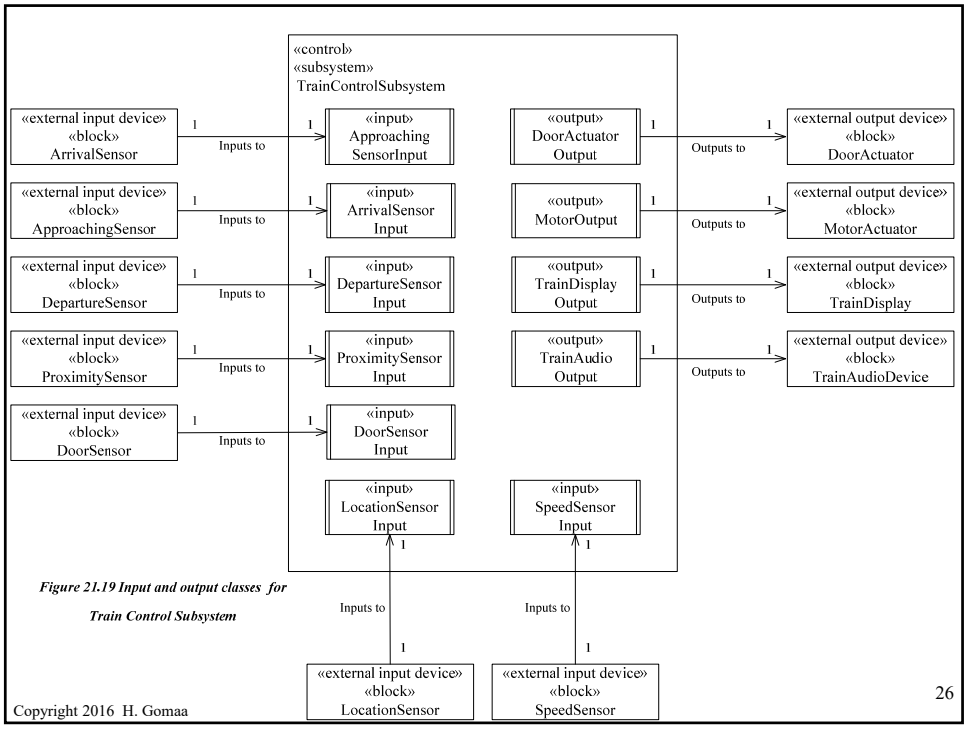
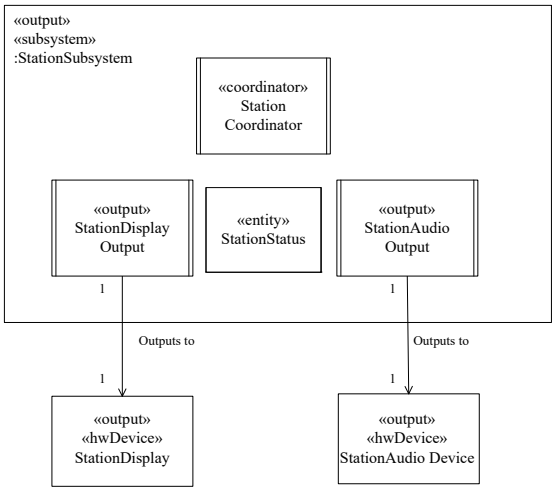


Figure 21.19 Input and output classes for Train Control Subsystem

Figure 21.20 Input and output classes for Station Subsystem



Dynamic Interaction Modeling of External Objects with Software System

- Starting with
  - Use case model
  - system context model
- From software system context diagram
  - Model software system as one aggregate object
  - Model instances of external objects from
- From use case model
  - Follow sequence of interactions described in use case
  - Depict interaction with external objects

Figure 21.21 Sequence diagram for Arrive at Station use case (external objects)

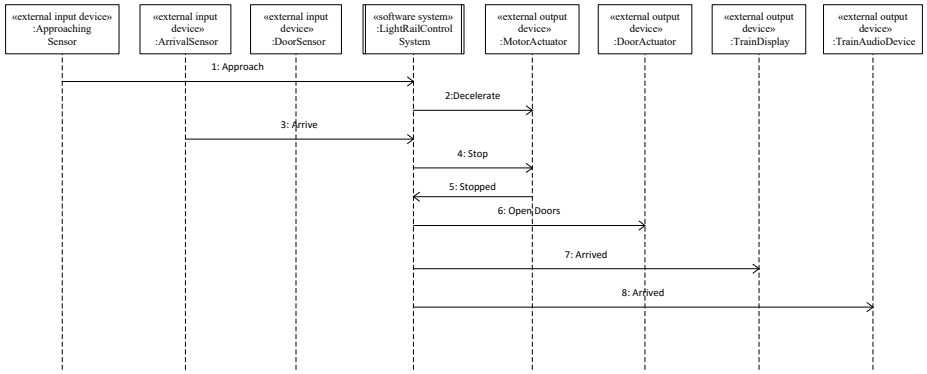


Figure 21.22 Sequence diagram for Arrive at Station use case (software objects)

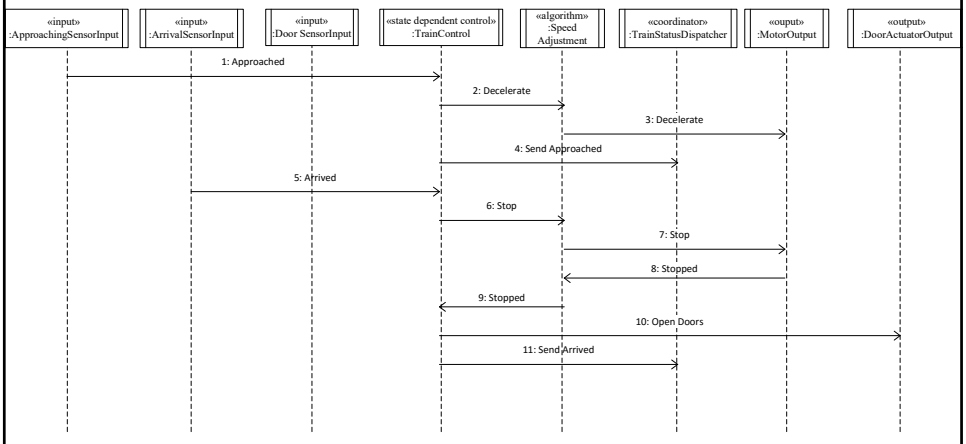


Figure 21.24 Sequence diagram for Control Train at Station use case

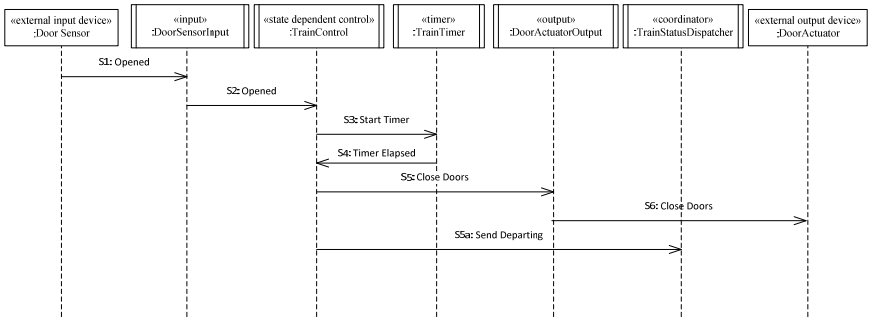


Figure 21.25 Sequence diagram for Depart from Station use case

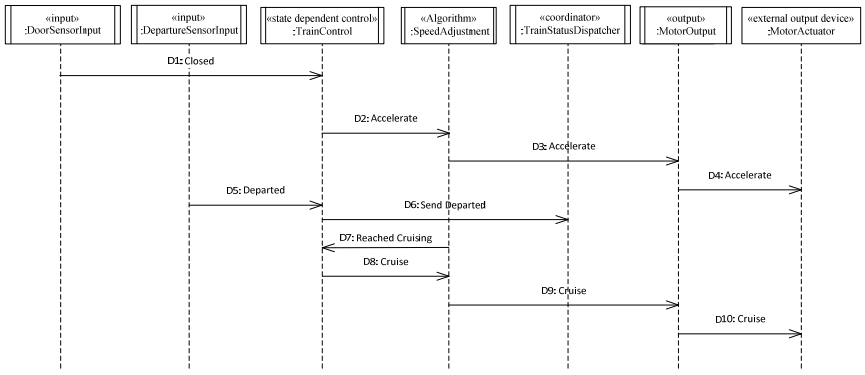
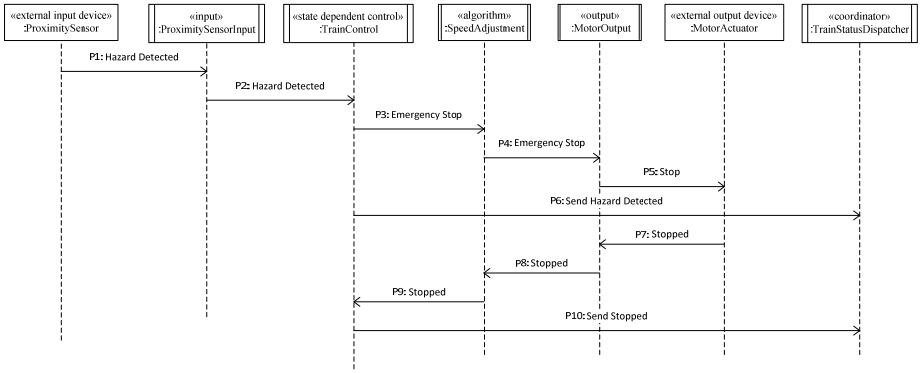
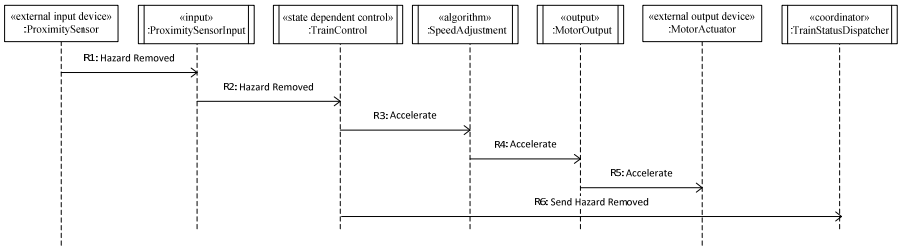


Figure 21.26 Sequence diagram for Detect Obstacle Presence use case



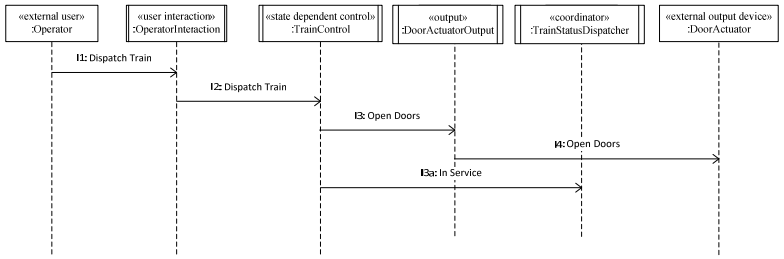
33

Figure 21.27 Sequence diagram for Detect Obstacle Removal use case



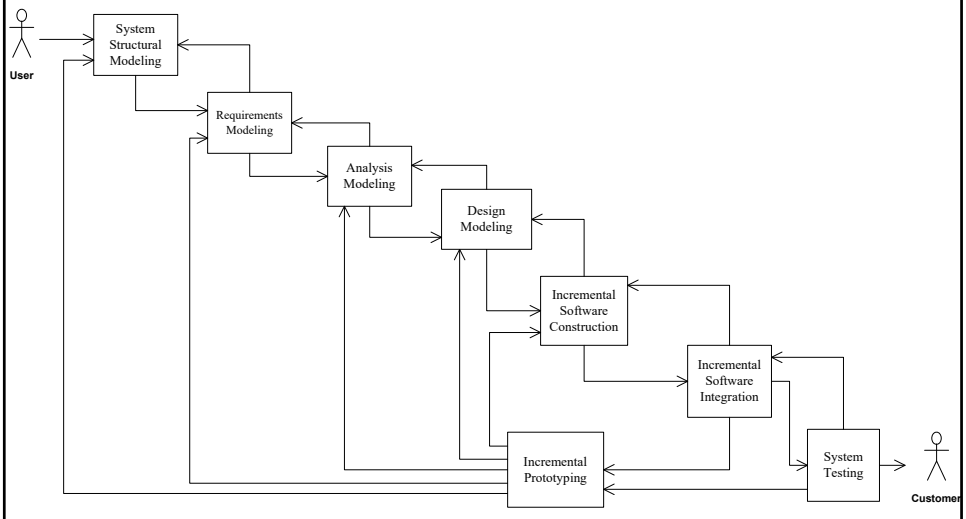
34

Figure 21.28 Sequence diagram for Dispatch Train use case



35

Figure 4.1 COMET/RTE life cycle model



36

Analysis Modeling

- Static Modeling
  - Define entity classes and relationships
- Dynamic State Machine Modeling
  - Real-time systems are highly state dependent
  - Actions depend on input event AND current sta**te**
- Object Structuring
  - Determine objects that realize each use case
- **Dynamic Interaction Modeling**
  - **Determine sequence of interactions among objects**

Figure 4.1 COMET/RTE life cycle model