SWE 760

Lecture 5:
Object Structuring for
Real-Time Embedded Systems

Hassan Gomaa
Department of Computer Science
George Mason University
Email: hgomaa@gmu.edu

References:

Copyright © 2016 Hassan Gomaa
All rights reserved. No part of this document may be reproduced in any form or by any means, without the prior written permission of the author.

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 state
• **Object Structuring**
  – **Determine objects that realize each use case**
• Dynamic Interaction Modeling
  – Determine sequence of interactions among objects

Object Structuring Criteria

• Determine all software objects in system
  – Use Object Structuring Criteria
  – Guidelines for identifying objects
• Structuring criteria depicted using stereotypes
  – **Stereotype** defines role of class or object in application
  – Class has same stereotype as objects instantiated from it
  – Depicted using guillemets
    • «entity», «boundary», «control»
• Objects are categorized
  – A **category** is a specifically defined division in a system of classification
Object Structuring Criteria

- Boundary objects
  - User interaction object
  - Device I/O object
  - Proxy object
- Entity objects
  - Long living objects that store information
  - Determined during static modeling
- Control objects
  - Decision making object
- Application Logic Objects
  - Encapsulates details of application

Figure 8.1: Classification of application classes by stereotype
Object Structuring in RT systems

- Concurrency is fundamental to RT systems
- During Object Structuring
  - Assume all objects are concurrent EXCEPT entity objects
  - Assume all communication between concurrent objects is asynchronous
- These initial decisions can be changed later during RT design

![Communicating Concurrent objects](image)

Object Structuring in RT systems

- During Object Structuring
  - Assume entity objects are passive
  - Assume all communication with entity object is synchronous (i.e., operation (method) invocation)

![Concurrent objects communicating with passive objects](image)
Object Structuring Criteria

- **Boundary objects**
  - Interface to and communicate with external environment
  - Each software boundary object interfaces to an external (real-world) object
    - User interaction object
      - Interfaces to and interacts with a human user
    - Device I/O object
      - Interfaces to I/O device
    - Proxy object
      - Interfaces to an external system
- Can determine boundary objects from *software system context diagram*
Object Structuring Criteria

- Control object
  - Coordinator object
    • Decision making object, not state dependent
    • Decides when, and in what order, other objects execute
  - State dependent control object
    • Defined by state machine
      – Statechart or state transition table
  - Timer object
    • Activated periodically

Object Structuring Criteria

- Application Logic Objects
  - Business Logic Object
    • For business (not RT) applications
  - Algorithm Object
    • Encapsulates algorithm used in problem domain
    • More usual in scientific, engineering, real-time domains
  - Service object
    • Provides a service for RT client objects
    • E.g., to store or retrieve data
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 state
- **Object Structuring**
  - **Determine objects that realize each use case**
- Dynamic Interaction Modeling
  - Determine sequence of interactions among objects