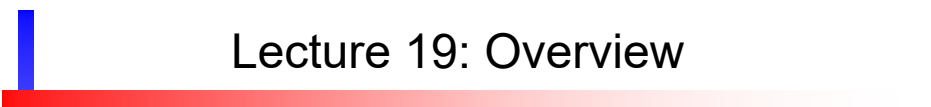


EECS 222: Embedded System Modeling Lecture 19

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine



Lecture 19: Overview

- Unified Modeling Language (UML)
 - Overview
 - Example Diagrams

Unified Modeling Language (UML)

- Goals
 - Raising the Level of Abstraction
 - Modeling of software applications
 - before coding!
 - Specification of software architecture
 - High-level description of software architecture to enable
 - scalability
 - security
 - robustness
 - maintenance
 - extendability
 - code reuse
 - Model Driven Architecture (MDA)
- Status
 - UML 2.0: Modeling Language in Software Engineering
 - standardized by OMG (Object Management Group) in 1997
 - standardized by ISO (Intl. Org. for Standardization) in 2005

EECS222: Embedded System Modeling, Lecture 19

(c) 2020 R. Doemer

3

Unified Modeling Language (UML)

- What is UML?
 - Graphical representation of ...
 - Software architecture
 - Software structure
 - Software behavior
 - Object relations
 - ...
 - 13 standard diagrams
 - Specification
 - Design
 - Documentation
 - Not executable!
 - Commercial tools available for ...
 - Graphical capture
 - Editing
 - Code generation (template code)

EECS222: Embedded System Modeling, Lecture 19

(c) 2020 R. Doemer

4

Unified Modeling Language (UML)

- UML Standard Diagrams
 - Structure Diagrams
 - Class Diagram
 - Object Diagram
 - Component Diagram
 - Composite Structure Diagram
 - Package Diagram
 - Deployment Diagram
 - Behavior Diagrams
 - Use Case Diagram
 - Activity Diagram
 - State Machine Diagram
 - Interaction Diagrams
 - Sequence Diagram
 - Communication Diagram
 - Timing Diagram
 - Interaction Overview Diagram

EECS222: Embedded System Modeling, Lecture 19

(c) 2020 R. Doemer

5

Unified Modeling Language (UML)

- UML Resources
 - Online Documents
 - Object Management Group (OMG)
 - www.uml.org
 - Online Tutorials
 - <https://www.tutorialspoint.com/uml/>
 - <http://www.sparxsystems.com/uml-tutorial.html>
 - Invited Talk at UCI in 2004
 - Dr. Wolfgang Mueller, C-LAB, Paderborn, Germany
 - Source of the following UML diagram examples

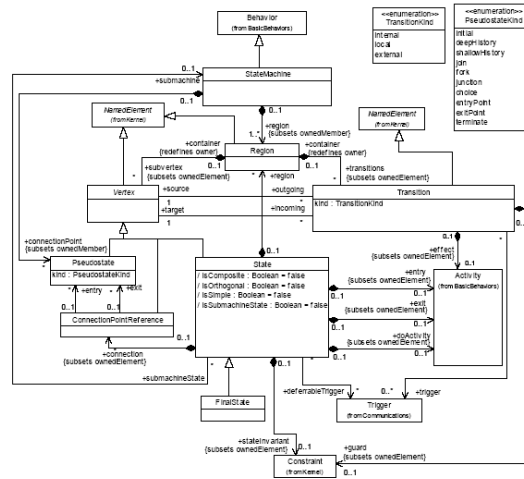
EECS222: Embedded System Modeling, Lecture 19

(c) 2020 R. Doemer

6

Unified Modeling Language (UML)

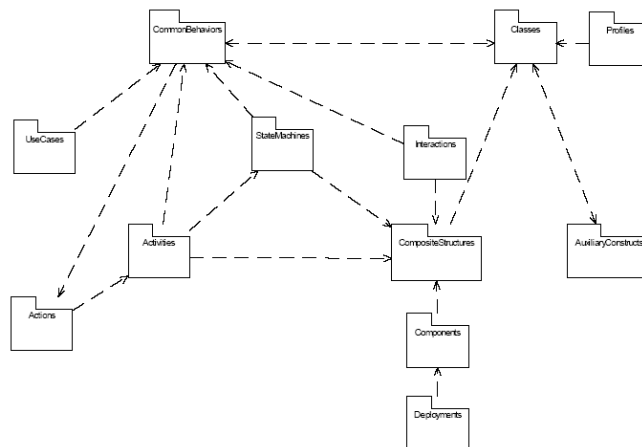
• Class Diagram Example



(source: W. Mueller)

Unified Modeling Language (UML)

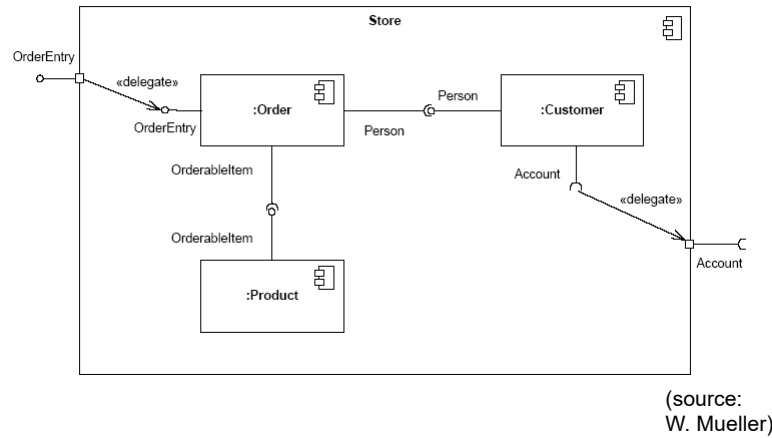
• Package Diagram Example



(source: W. Mueller)

Unified Modeling Language (UML)

- Component Diagram Example



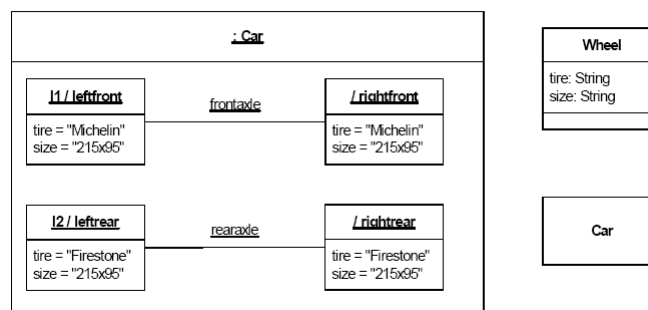
EECS222: Embedded System Modeling, Lecture 19

(c) 2020 R. Doemer

9

Unified Modeling Language (UML)

- Composite Structure Diagram Example



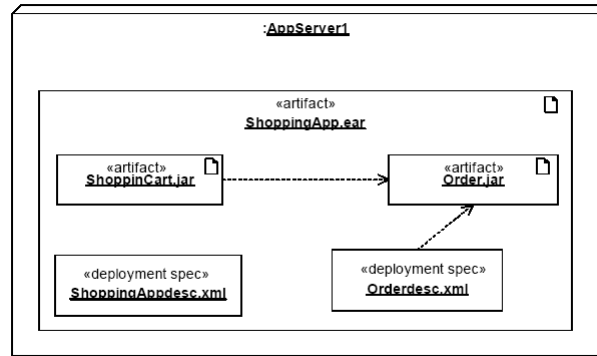
EECS222: Embedded System Modeling, Lecture 19

(c) 2020 R. Doemer

10

Unified Modeling Language (UML)

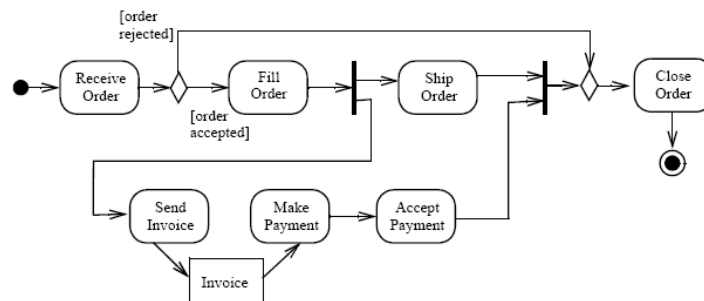
- Deployment Diagram Example



(source: W. Mueller)

Unified Modeling Language (UML)

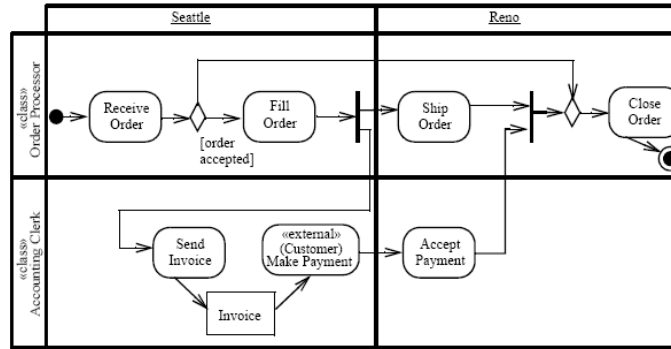
- Activity Diagram Example



(source: W. Mueller)

Unified Modeling Language (UML)

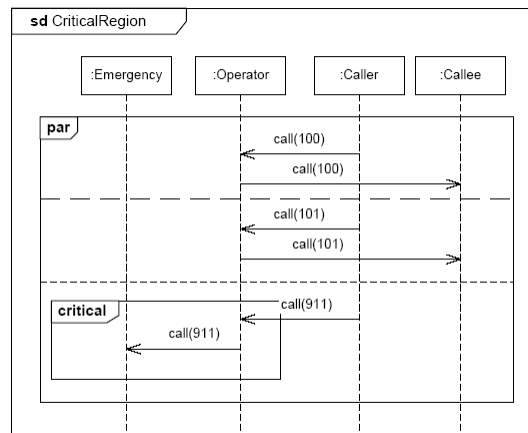
- Activity Diagram Example with “swim lanes”



(source: W. Mueller)

Unified Modeling Language (UML)

- Sequence Diagram Example



(source: W. Mueller)

Unified Modeling Language (UML)

- Use Case Diagram Examples

The diagram shows a system boundary labeled "Telephone Catalog" containing four use cases: "Check Status", "Place Order", "Fill Orders", and "Establish Credit". Three actors are shown: "Customer", "Salesperson", and "Shipping Clerk". "Customer" is connected to "Check Status", "Place Order", and "Fill Orders". "Salesperson" is connected to "Check Status". "Shipping Clerk" is connected to "Fill Orders". "Supervisor" is connected to "Establish Credit". Labels "use case", "actor", and "subject" point to their respective symbols.

(source: W. Mueller)

EECS222: Embedded System Modeling, Lecture 19 (c) 2020 R. Doemer 15

Unified Modeling Language (UML)

- State Machine Diagram Examples

The slide contains three state machine diagrams. The first is a state transition diagram with states State0, State1, State2, State3, and State4. Transitions are labeled with guards: State0 to State2 ([a < 0]), State0 to State3 ([a = 5]), State0 to State4 ([a > 7]), State1 to State0 (e2[b < 0]), and State1 to State0 (e1[b < 0]). The second diagram shows a "Process" state with a fork, two parallel paths (A1 to A2 and B1 to B2), a join, and a "Cleanup" state. The third diagram, titled "Dialing", shows a state transition from "Start" to "Partial Dial" on the event "digit(n)". The "Start" state has entry, start dial tone, and exit/stop dial tone actions. The "Partial Dial" state has an entry/number.append(n) action and a self-loop on "digit(n)". A guard [number.isValid()] leads to a final state.

(source: W. Mueller)

EECS222: Embedded System Modeling, Lecture 19 (c) 2020 R. Doemer 16