

EEE499 – Model-driven Development of Real-Time Systems

An Introduction to Model-Driven
Software Development to the course

ROYAL MILITARY COLLEGE OF CANADA
ELECTRICAL & COMPUTER
ENGINEERING



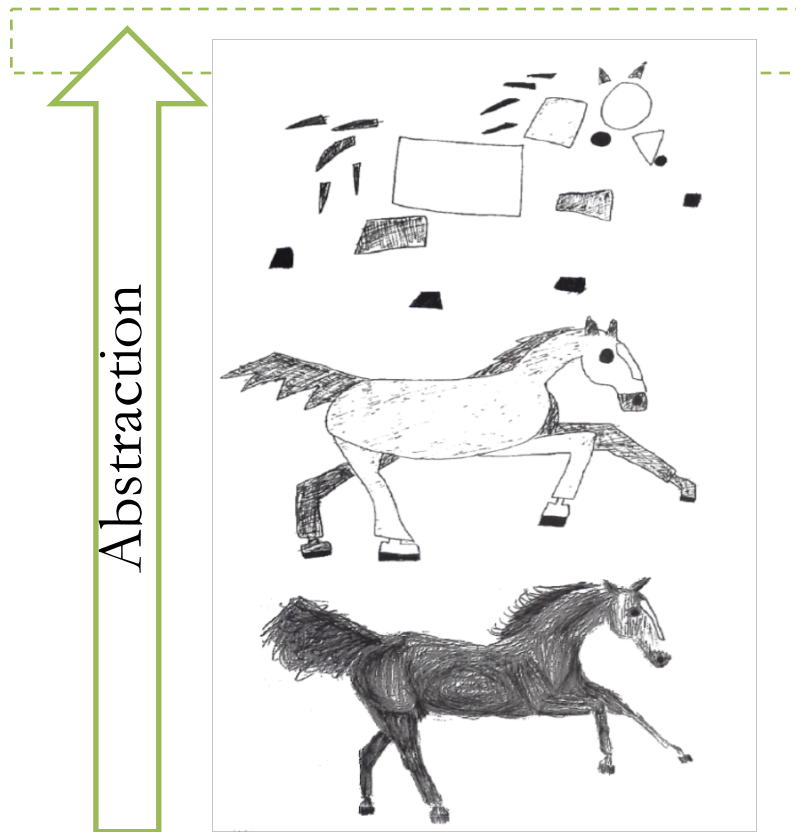
GÉNIE ÉLECTRIQUE
ET GÉNIE INFORMATIQUE
COLLÈGE MILITAIRE ROYAL DU CANADA



Acknowledgement

The original material for this section was developed by [Prof. Juergen Dingel](#) (Queen's University)

Model Driven Development



Automation



Analysis

Expressing SW models: Overview

Examples of software modeling languages

1. **UML** (for modeling everything)
 - **language**: collection of 14 diagram types
 - **analysis**: e.g., well-formedness, approaches to consistency, reachability
2. **UML-RT (for soft real-time embedded)**
 - **language**: much smaller, domain-specific subset of UML
3. **Stateflow/Simulink (for control systems)**
 - **language**: domain-specific combination of statemachines and dataflow
4. **SMV, Promela** (for concurrent systems)
 - **language**: concurrent, imperative language with message passing
 - **analysis**: temporal logic model checking (i.e., exhaustive state space exploration) using NuSMV, Spin

Lots more:

Petri nets, queuing networks, synchronous languages, ...

Modeling Languages

Modelica

- Physical systems
- Equation-based

Simulink

- Continuous control, DSP
- time-triggered dataflow

Stateflow

- Reactive systems
- Discrete control
- State-machine-based

Lustre/SCADE

- Embedded real-time
- Synchronous dataflow

UML-RT

- Embedded, real-time
- State-machine-based

Examples in
[Voe13, Kel08]

EGGG
[Orw00]

AADL

- Embedded, real-time

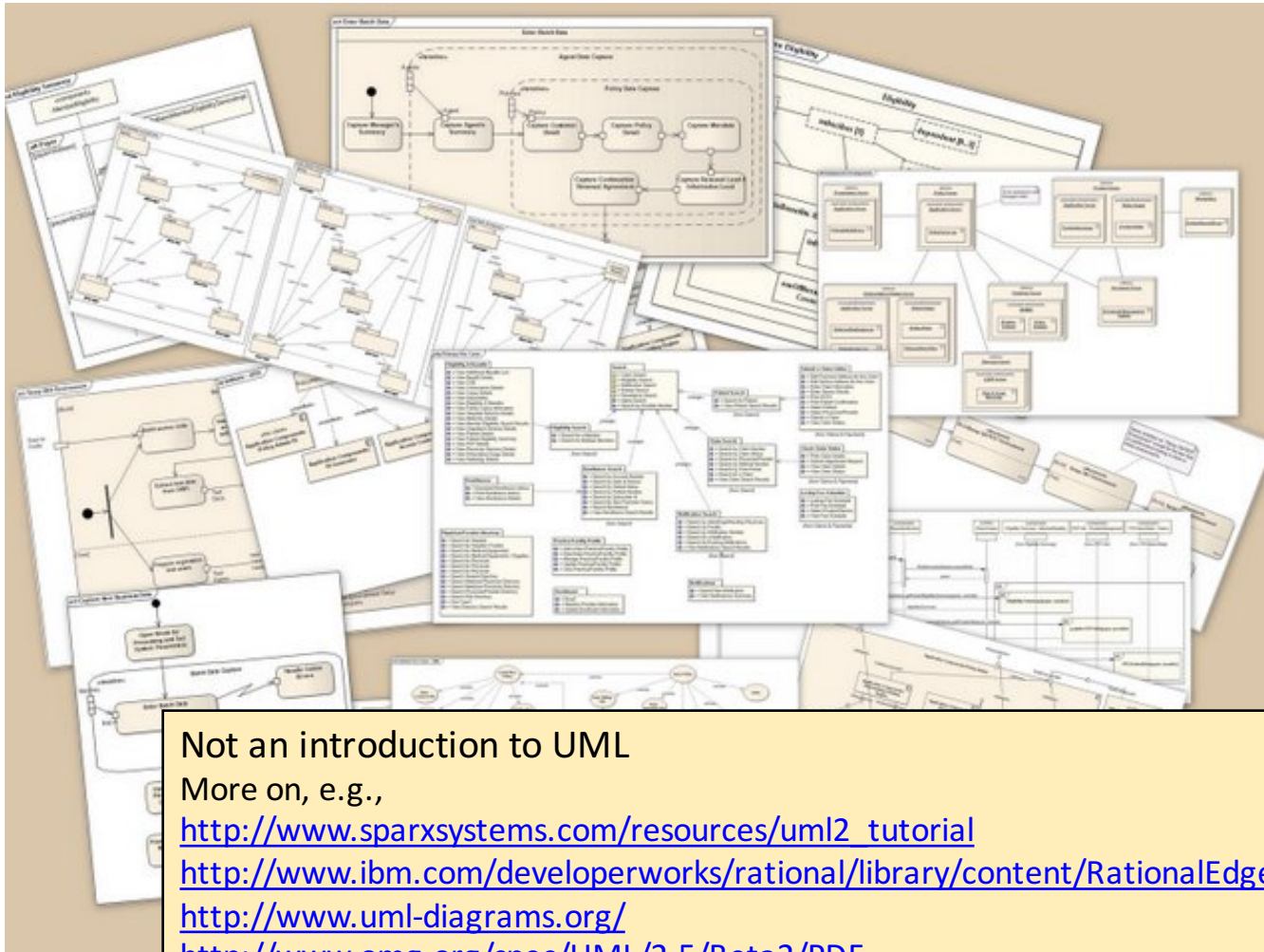
UML

UML MARTE

- Embedded, real-time

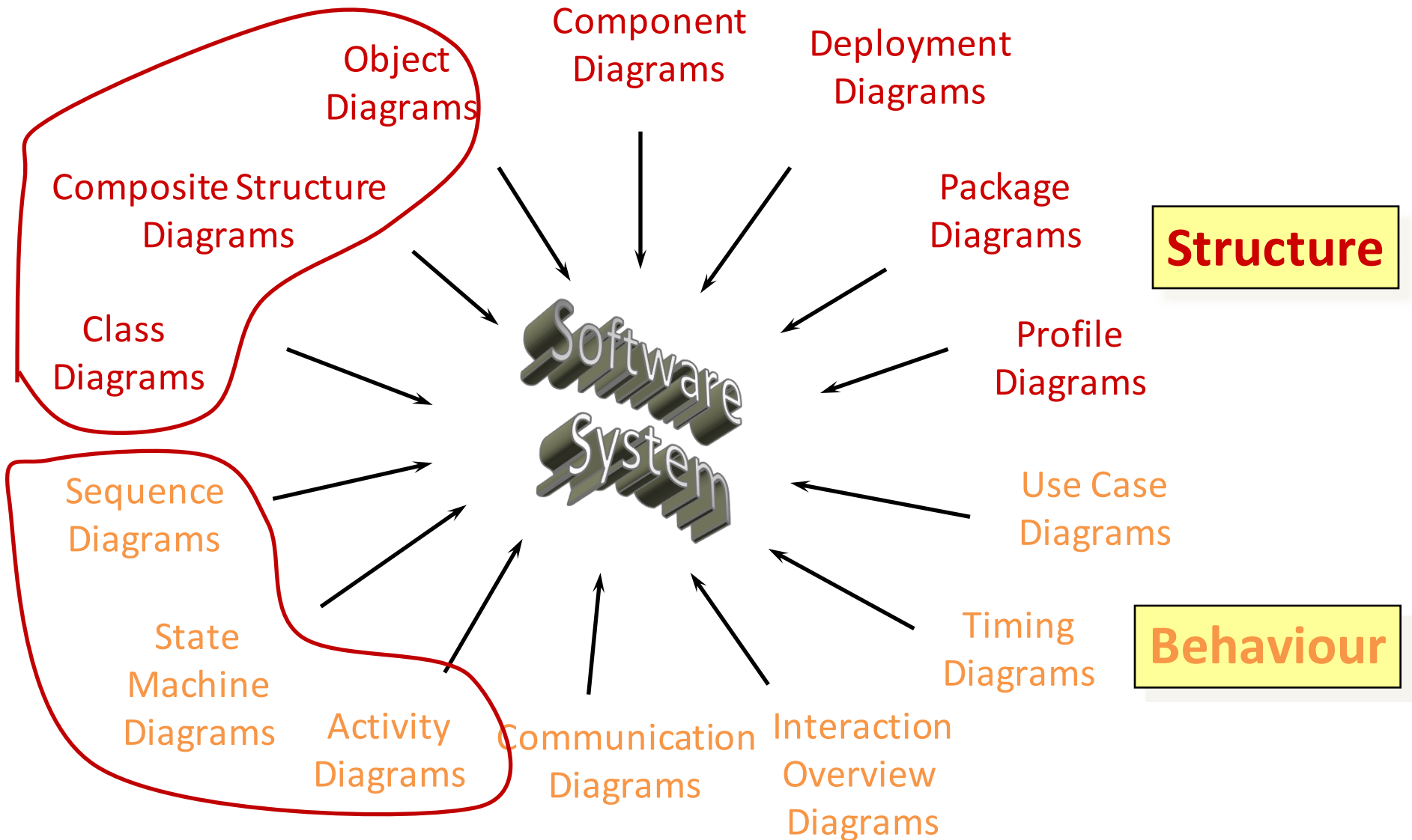


UML: A brief overview

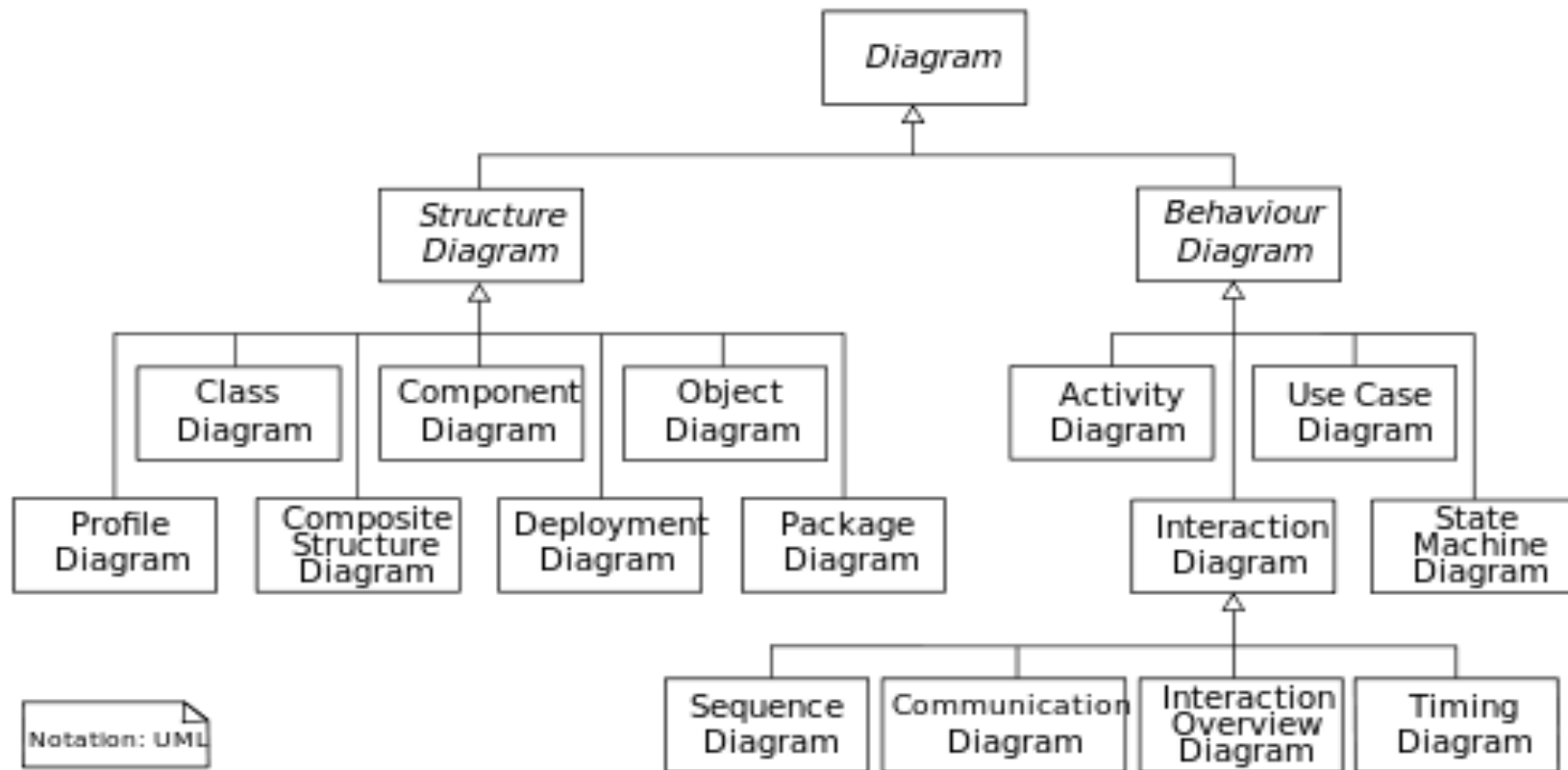


Not an introduction to UML
More on, e.g.,
http://www.sparxsystems.com/resources/uml2_tutorial
<http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/>
<http://www.uml-diagrams.org/>
<http://www.omg.org/spec/UML/2.5/Beta2/PDF>
http://link.springer.com/chapter/10.1007%2F978-3-642-30982-3_1

UML: 14 Different Diagram Types

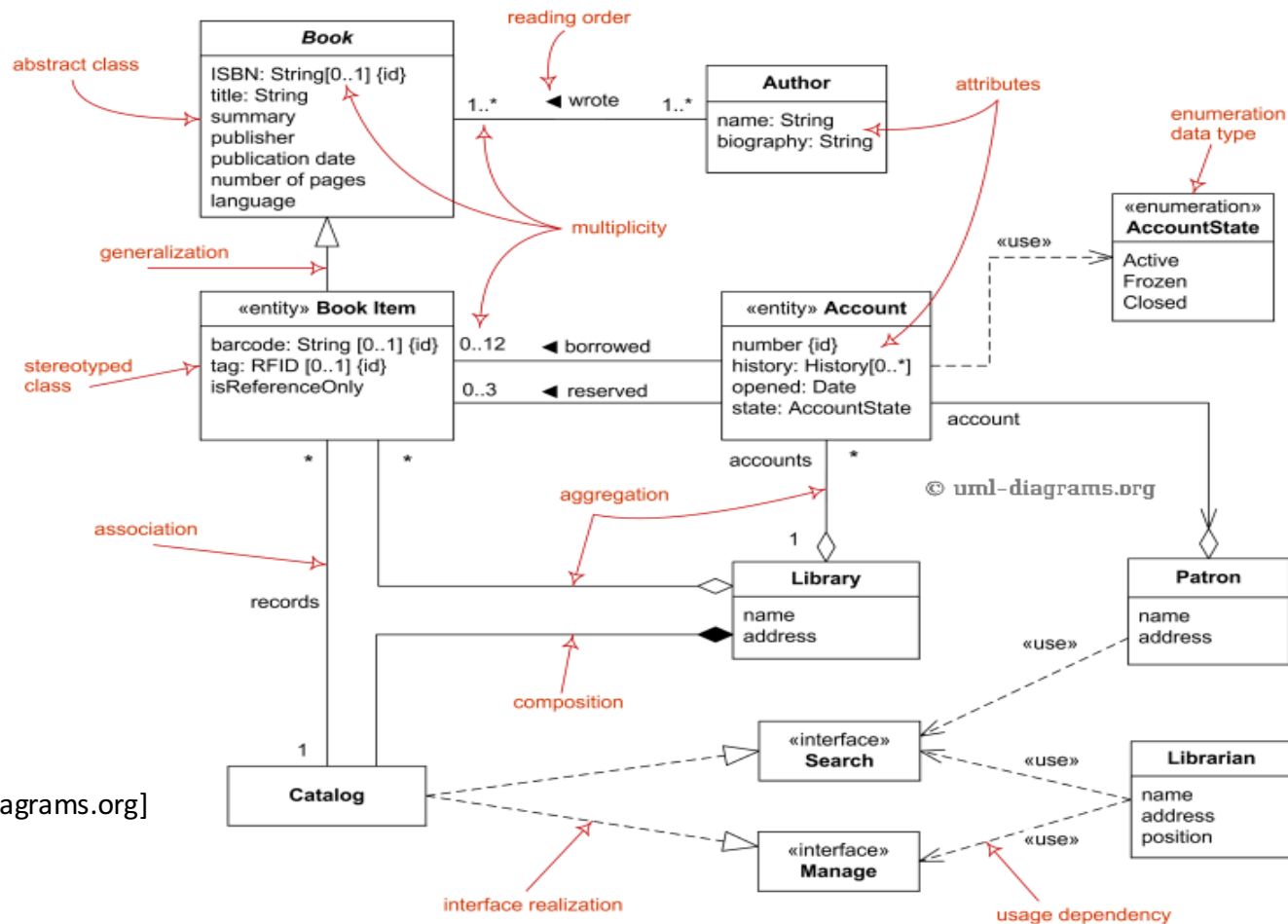


UML: 14 Different Diagram Types (Cont'd)



UML: Class Diagrams (Cont'd)

Shows classes/concepts, their attributes, operations & relationships

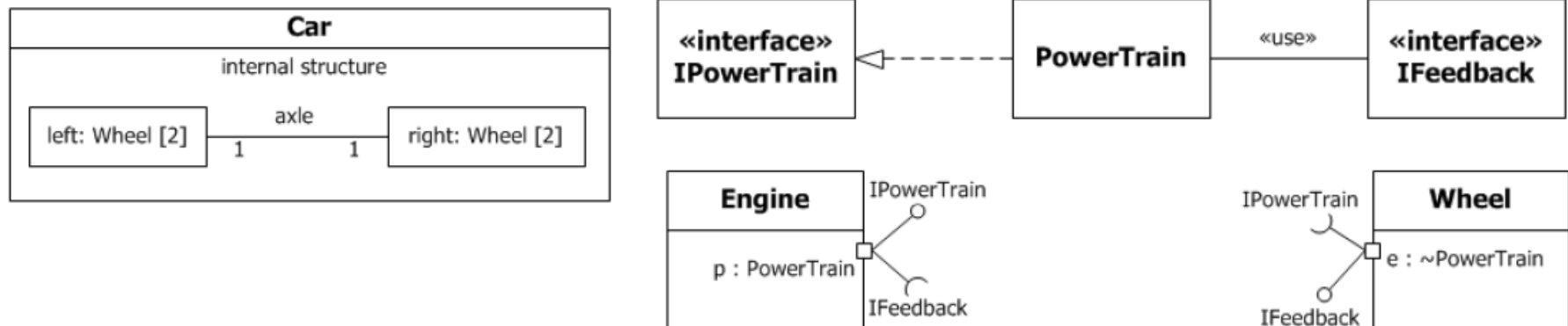


[www.uml-diagrams.org]

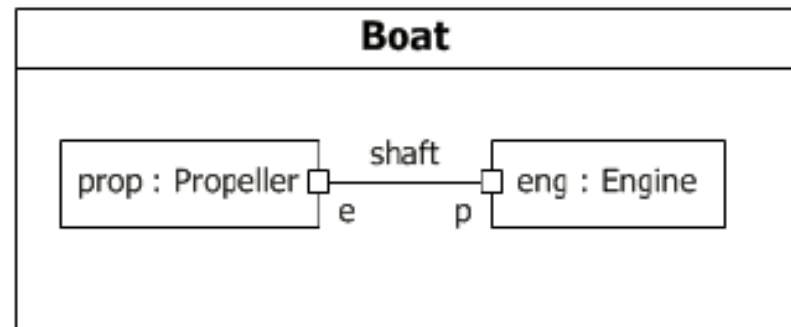
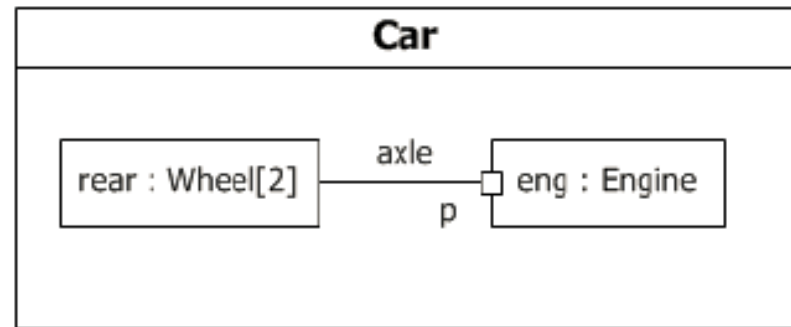
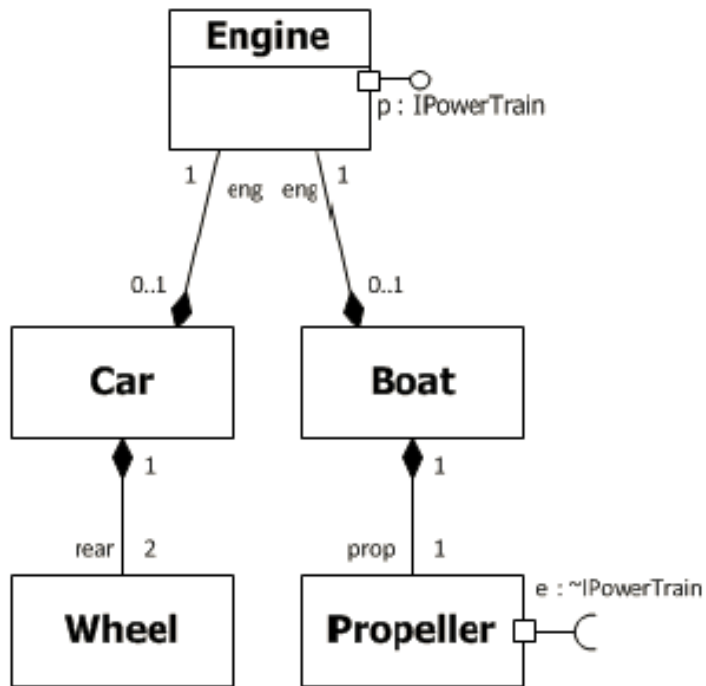
UML

UML: Composite Structure Diagrams

- Shows internal structure of StructuredClassifier, including interaction points to other parts of system
- **Key concepts**
 - **Part:** Properties specifying instances that StructuredClassifier owns (i.e., properties w/ aggregationKind=composite)
 - **Port:** typed element defining interaction between classifier and environment; may specify provided and required services (via **interfaces**)
 - **Connector:** specifies links between parts; typically represents possibility to communicate; typed by Association

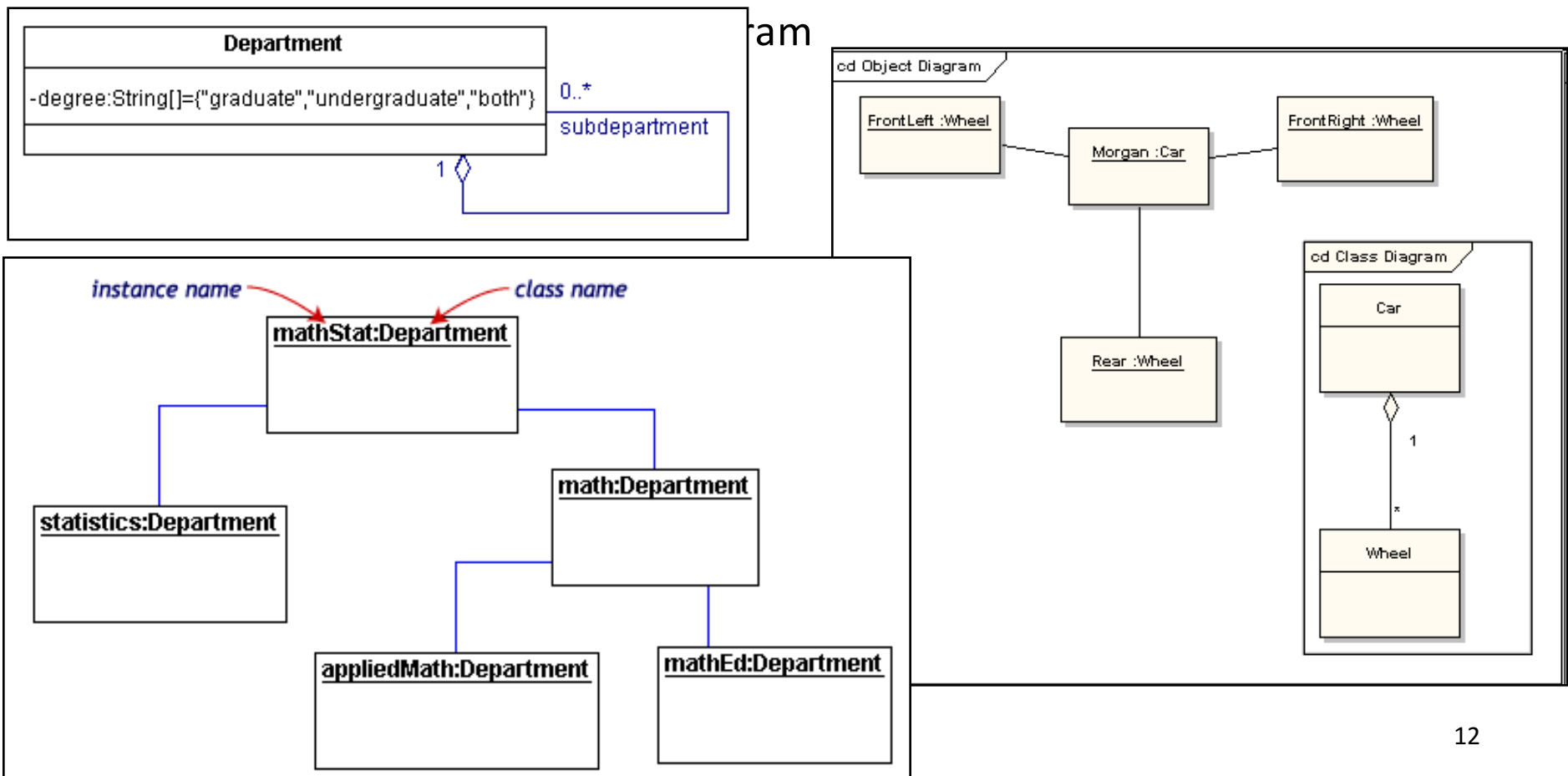


UML: Composite Structure Diagrams (Cont'd)



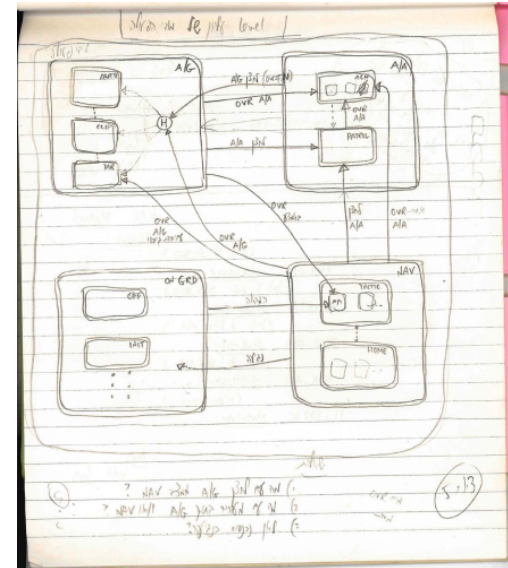
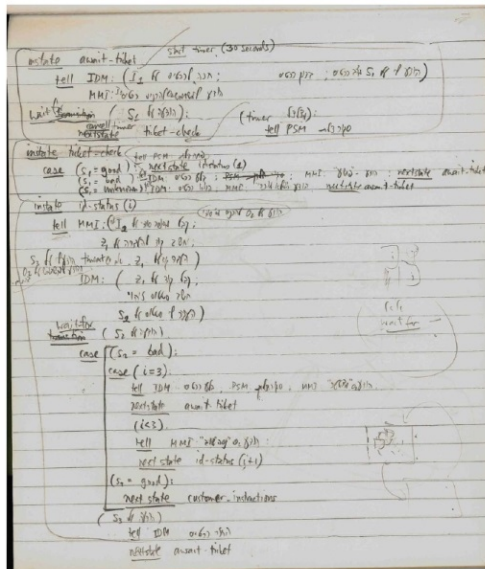
UML: Object Diagrams

- Shows objects/instances and their relationships at particular point in time (a.k.a., “snapshot” or “state”)



UML: State Machines

David Harel



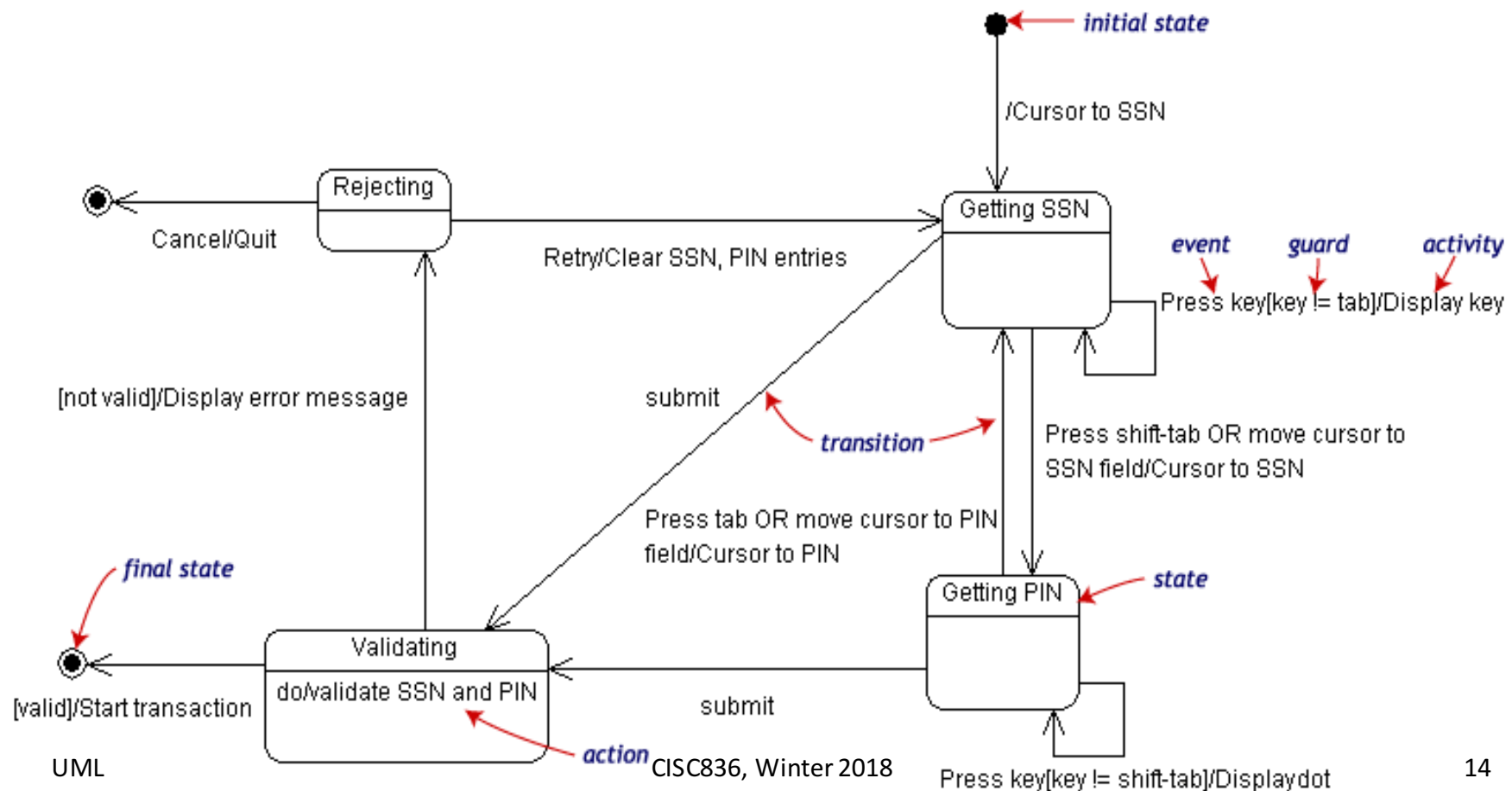
“The pictures were simply doing a much better job of setting down on paper the system's behavior, as understood by the engineers, and we found ourselves discussing the avionics and arguing about them over the diagrams, not the statocols.”

[Har07]

[Har07] D. Harel. Statecharts in the Making: A Personal Account. 3rd ACM SIGPLAN Conference on History of Programming Languages. 2007.

UML: State Machine Diagrams

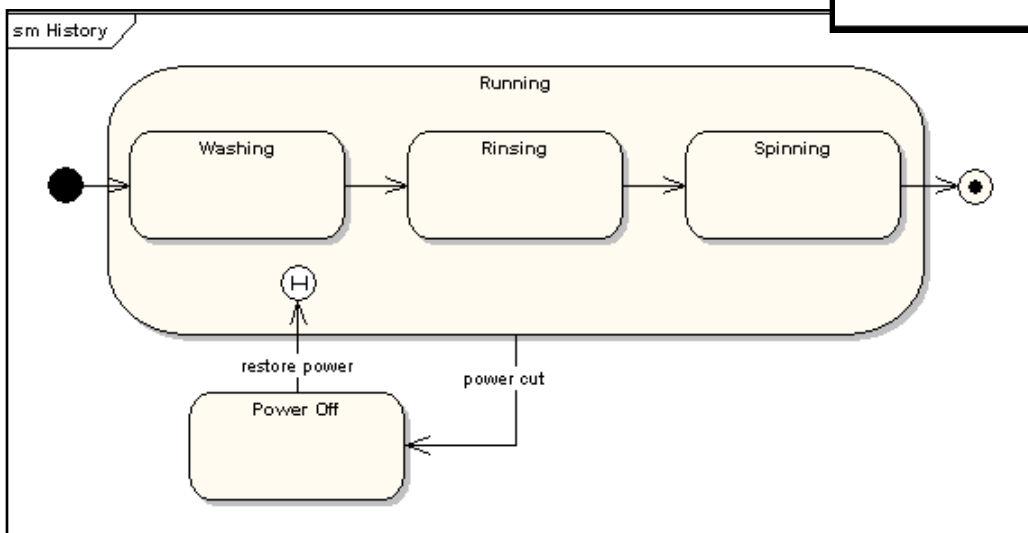
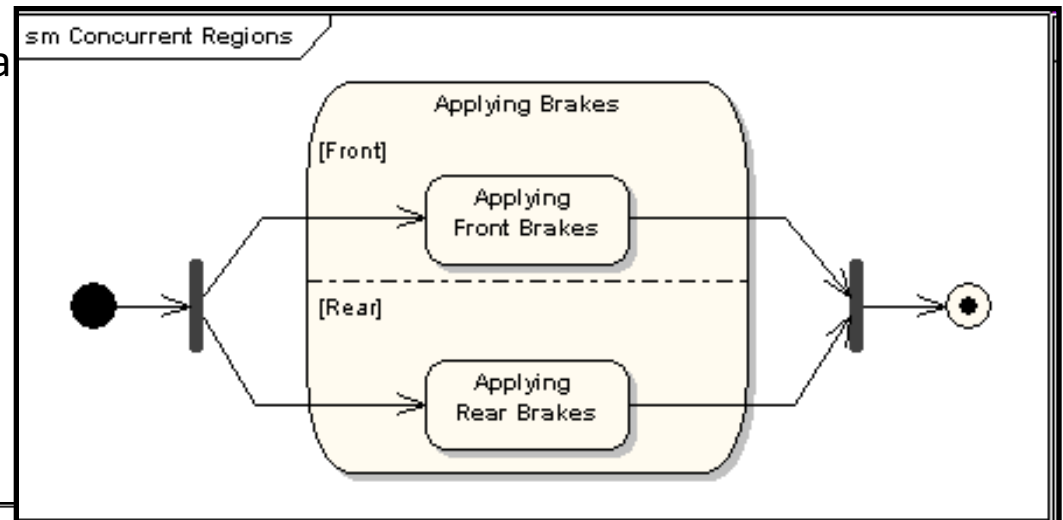
Show behaviour as sequences of state changes caused by transitions triggered by events

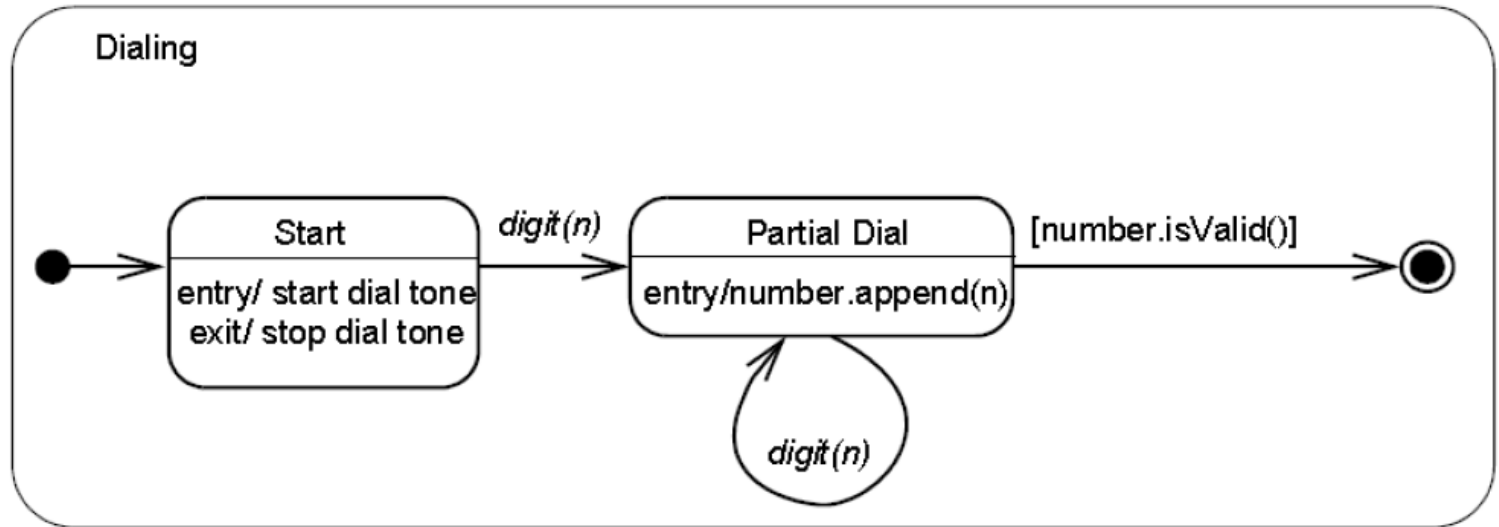


UML: State Machine Diagrams (Cont'd)

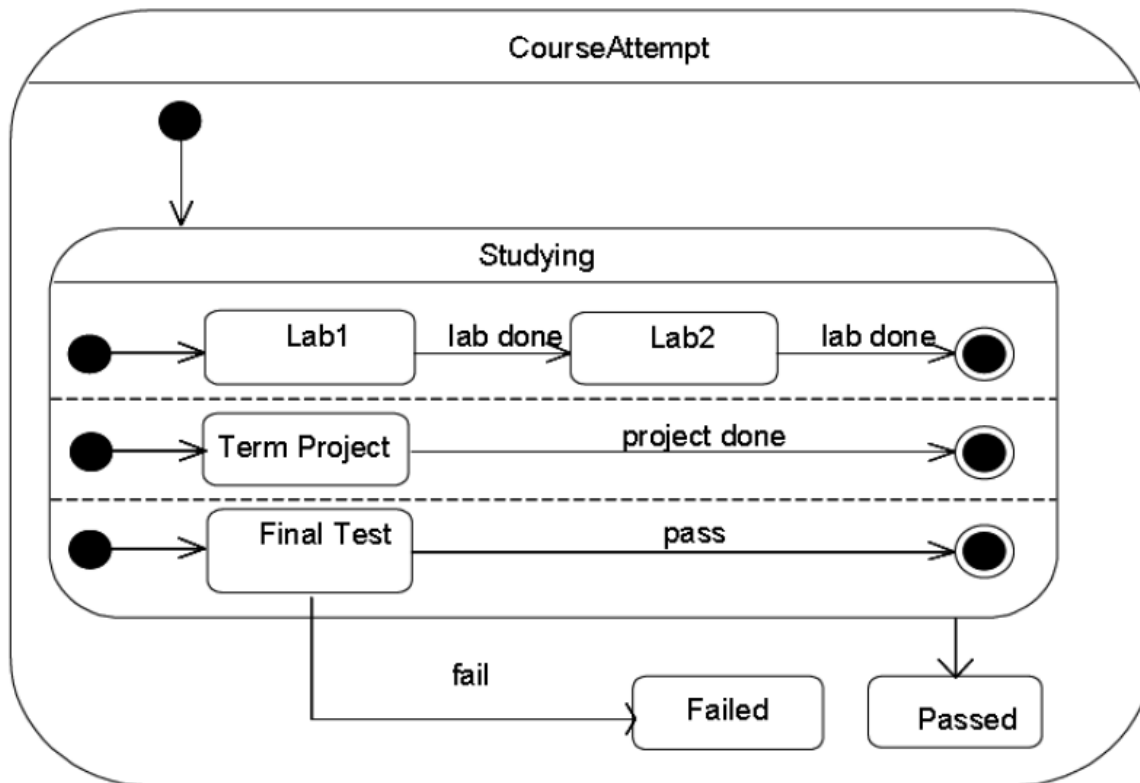
■ Features

- Composite states (hierarchical)
 - Group transitions
- History states
- (Orthogonal, concurrent) regions (and-states)
- Entry, exit, do-actions

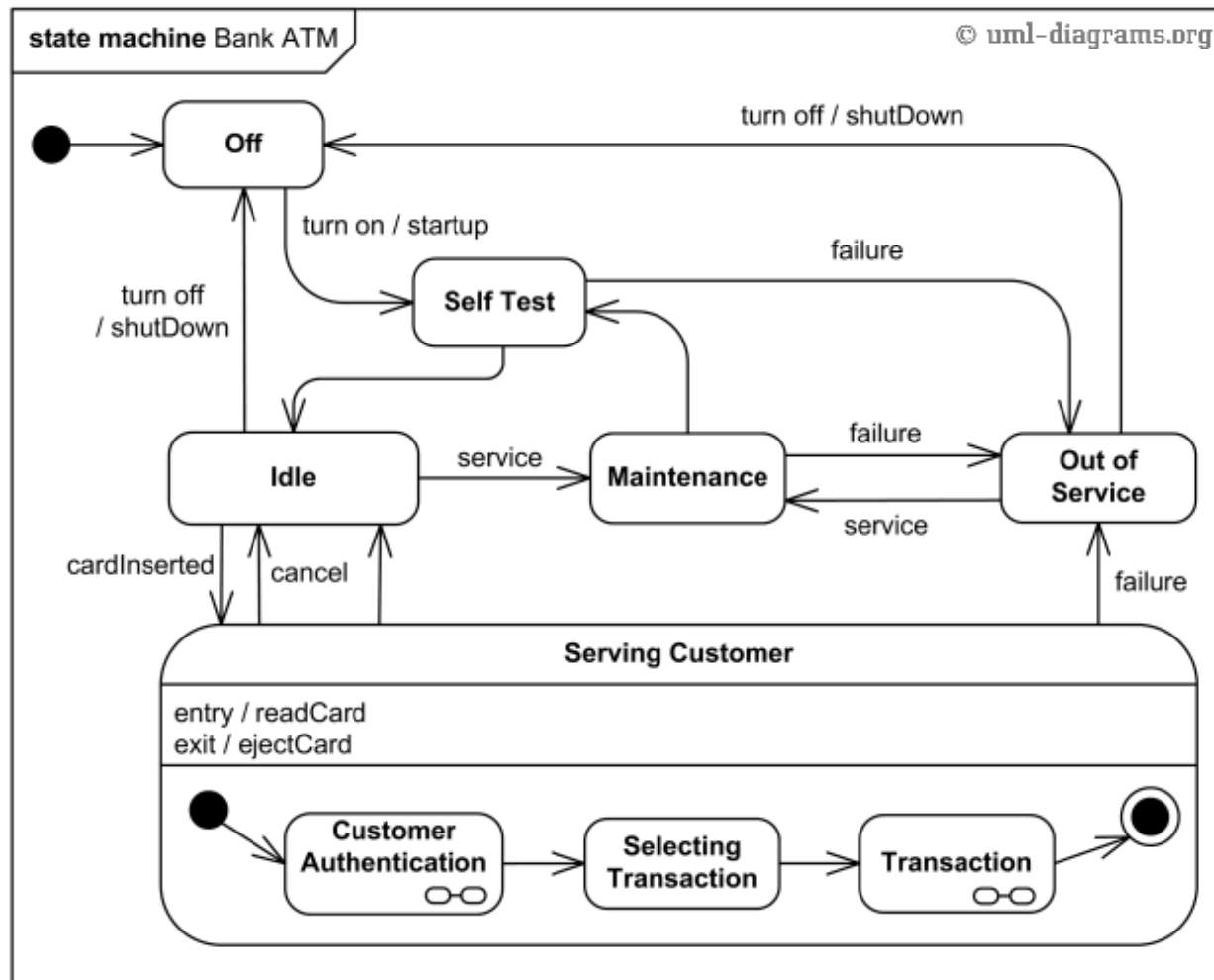




UML: State Machine Diagrams (Cont'd)



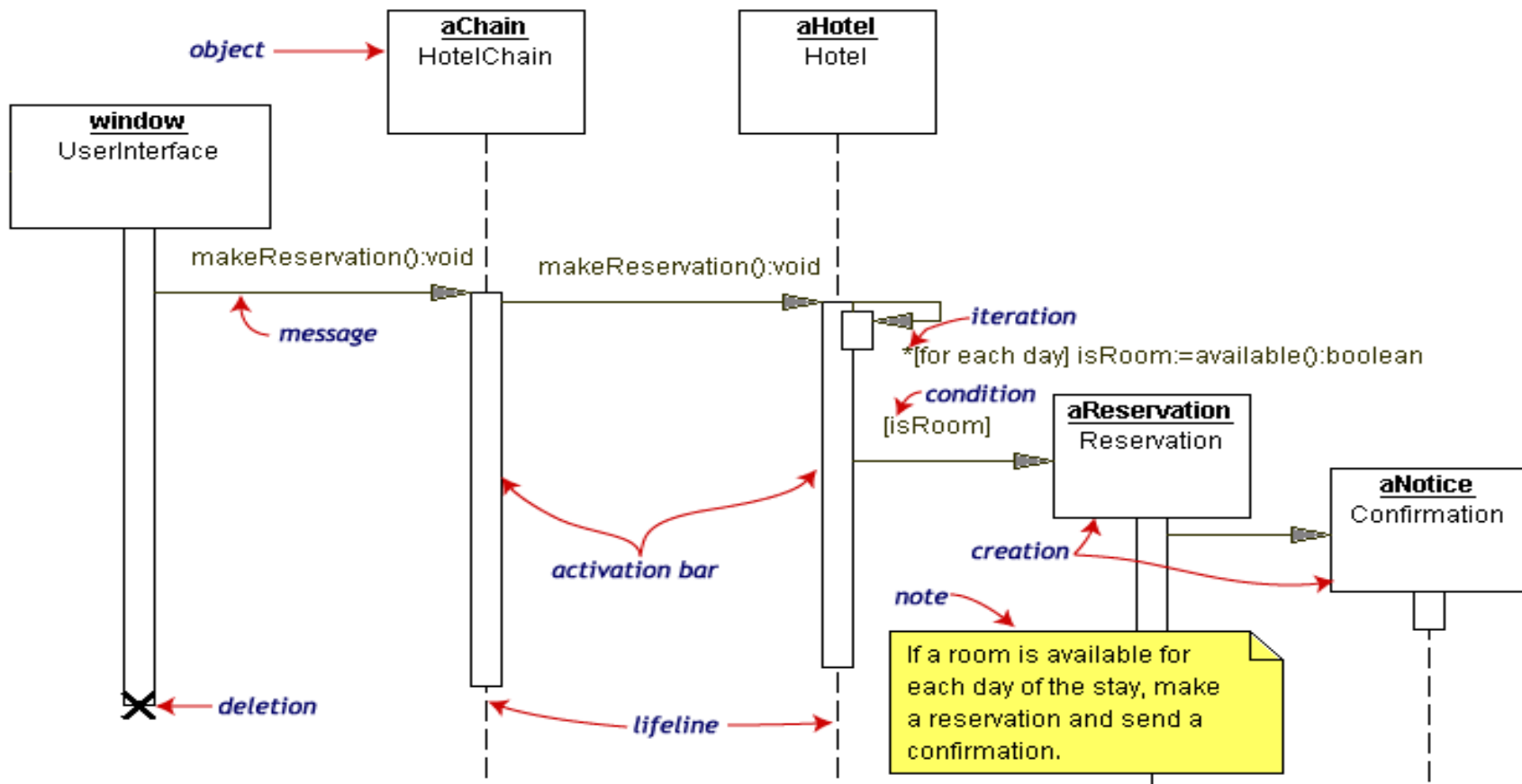
UML: State Machines (Cont'd)



[www.uml-diagrams.org]

UML: Sequence Diagrams

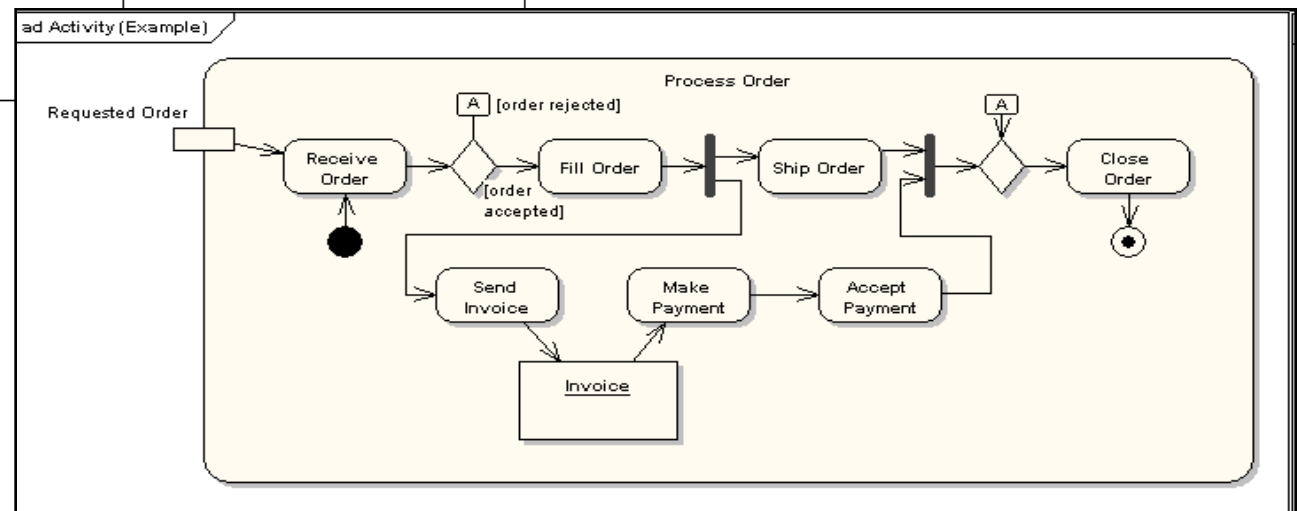
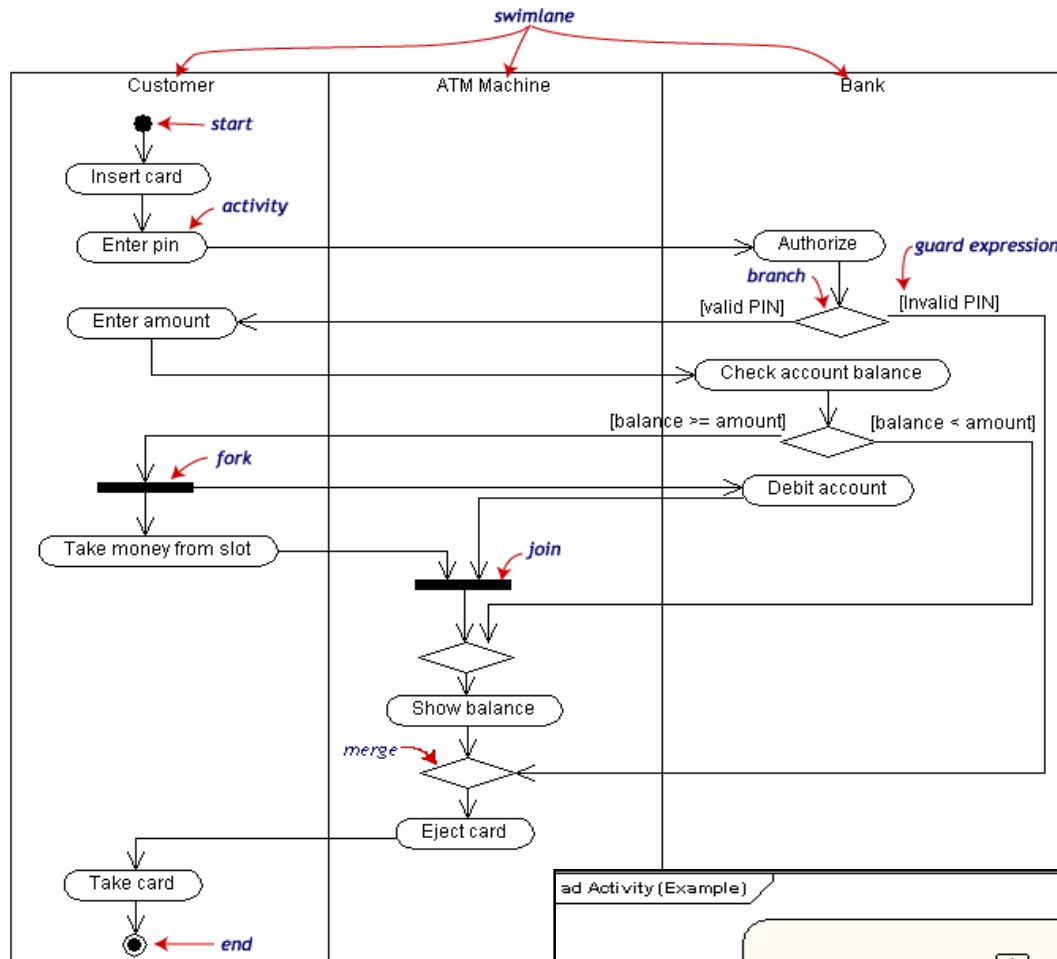
- Show behaviours as sequences of messages b/w objects



UML: Activity Diagrams

- Show behaviours as sequences of activities
- **Features**
 - **Two kinds of flow:** control and data
 - **Different kinds of behaviour invocation:** synchronous, asynchronous
 - **Different kinds of control nodes:** initial, final, fork, join, decision, merge
 - **Different composition mechanisms:** loops, conditionals, interruptible regions, exceptions
 - **Structuring mechanisms:** partitions, swimlanes
 - **Support for data flow:** edge weights, multiplicities on pins
- **Semantics:** Petri net-based “token/offer” semantics with deadlock avoidance rules

UML: Activity Diagrams (Cont'd)



UML: Tools

- **Commercial**
 - Rational RSA (IBM)
 - Rhapsody (IBM)
 - MagicDraw (No Magic)
- **Open source**
 - Papyrus
 - eclipse.org/papyrus
 - Papyrus for Information Modeling (for class diagrams)
 - https://wiki.eclipse.org/Papyrus_for_Information_Modeling
 - Mentor Graphics xtUML
 - <http://www.xtuml.org/>
 - USE (for OCL)
 - sourceforge.net/apps/mediawiki/useocl
- **Web-based**
 - Draw.io

UML: Summary

- De facto standard in software modeling
- Rich “dictionary” of model concepts
 - UML 2.5 Spec has 809 pages
 - “*UML was designed to be used selectively*”
Selic in [Pet14] Bran
 -) best to approach study of UML with particular purpose, need
- Tool support
 - Still a problem, but getting better
 - Increasingly open source