TTool Training

I. Introduction to UML

Ludovic Apvrille

ludovic.apvrille@telecom-paris.fr

Eurecom, Office 223

Outline of the Training

TELECOM PARIS
école nationale supérieure des télécommunications

- **Introduction to UML**
  - ***Modeling with UML***
  - ***Main diagrams for embedded systems and protocols***
- **The TURTLE profile**
  - ***Design with TURTLE***
  - ***Analysis with TURTLE***
- **The TURTLE Toolkit (TTool)**
- **Exercises**

- **Special thanks to Pierre de Saqui-Sannes who actively participated in the elaboration of these slides**

I. Introduction to UML

TELECOM PARIS
école nationale supérieure des télécommunications

- Introduction to modeling
- OMG
- UML 1.5 and UML 2.0
- UML for embedded systems and protocols
- Objects in a nutshell
- Analysis with UML
- Design with UML
- Detailed design with UML

What is UML?

TELECOM PARIS
école nationale supérieure des télécommunications

- **UML = Unified Modeling Language**

- **Main characteristics of UML**
  - ***Graphical modeling language for complex systems***
    - **Specification, design, automatic code generation, documentation**
    - **Independent of any programming language**
  - ***Object-oriented design***
  - ***Supported by many CASE Tools***
    - **CASE = Computer-Aided Software Engineering**
  - ***Warning: no standard UML methodology***

A Few Questions …

- What is graphical modeling?
- Where does UML come from?
- Why should engineers use UML?
- Is the use of UML relevant for embedded systems / real-time systems?

What is Modeling?

TELECOM PARIS
école nationale supérieure des télécommunications

- **A modeling = an abstraction of the system to design**
  - ***Representation of the main functionalities of a complex system***
  - ***Non relevant details are ignored***
- **Abstractions make it possible to deal with complexity**
  - ***An engineer, or a development team, cannot have a global understanding of complex systems***
- **A modeling is a view of a system according to some assumptions**

Who Uses Modeling?

TELECOM PARIS
école nationale supérieure des télécommunications

- **Architects**
- **Tailors**
- **Statisticians**
- **Engineers**
  - ***Mechanics, Mechanics of fluid,***
  - ***Protocols,***
  - ***Electronic, microelectronic***
- **No exception for software!**
  - ***And more particularly, for embedded systems***

Why Use UML for Modeling?

TELECOM PARIS
école nationale supérieure des télécommunications

- **Standard notation**
  - ***Known by a growing number of people***
  - ***Supported by matured tools***
- **Best understanding of systems by**
  - ***Clients, experts of the domain, designers, programmers***
- **Support of engineering work**
  - ***Abstract view of the system***
  - ***Life cycle***
    - **Focused on first steps: requirement analysis, design**
    - **Simulation, automatic generation of code (ADA, C, Java, C++, etc.)**
    - **Documentation**
    - **Maintenance, revision**
  - ***Reuse***

Gathering on UML

TELECOM PARIS
école nationale supérieure des télécommunications

- **UML gathers best practices of software engineering**
- **Modeling of complex (and software-based) systems**
- **OMG (Object Management Group) standard**
  - ***The reference***
  - ***http://www.uml.org***
  - ***A notation***
    - **Box semantics**
- **12 diagrams for expressing complementary point of views**

Gathering on UML (Cont.)

TELECOM PARIS
école nationale supérieure des télécommunications

- **A notation**
  - ***Semantics?***
    - **Metamodel**
    - **No formal semantics**
  - ***No methodology***
    - **Process suggested by UML tool dealers**
      - ***Unified process***
        - RUP – Rational Unified Process
  - ***Extension capabilities***
    - **Profiles**

UML Views and Diagrams

TELECOM PARIS
école nationale supérieure des télécommunications

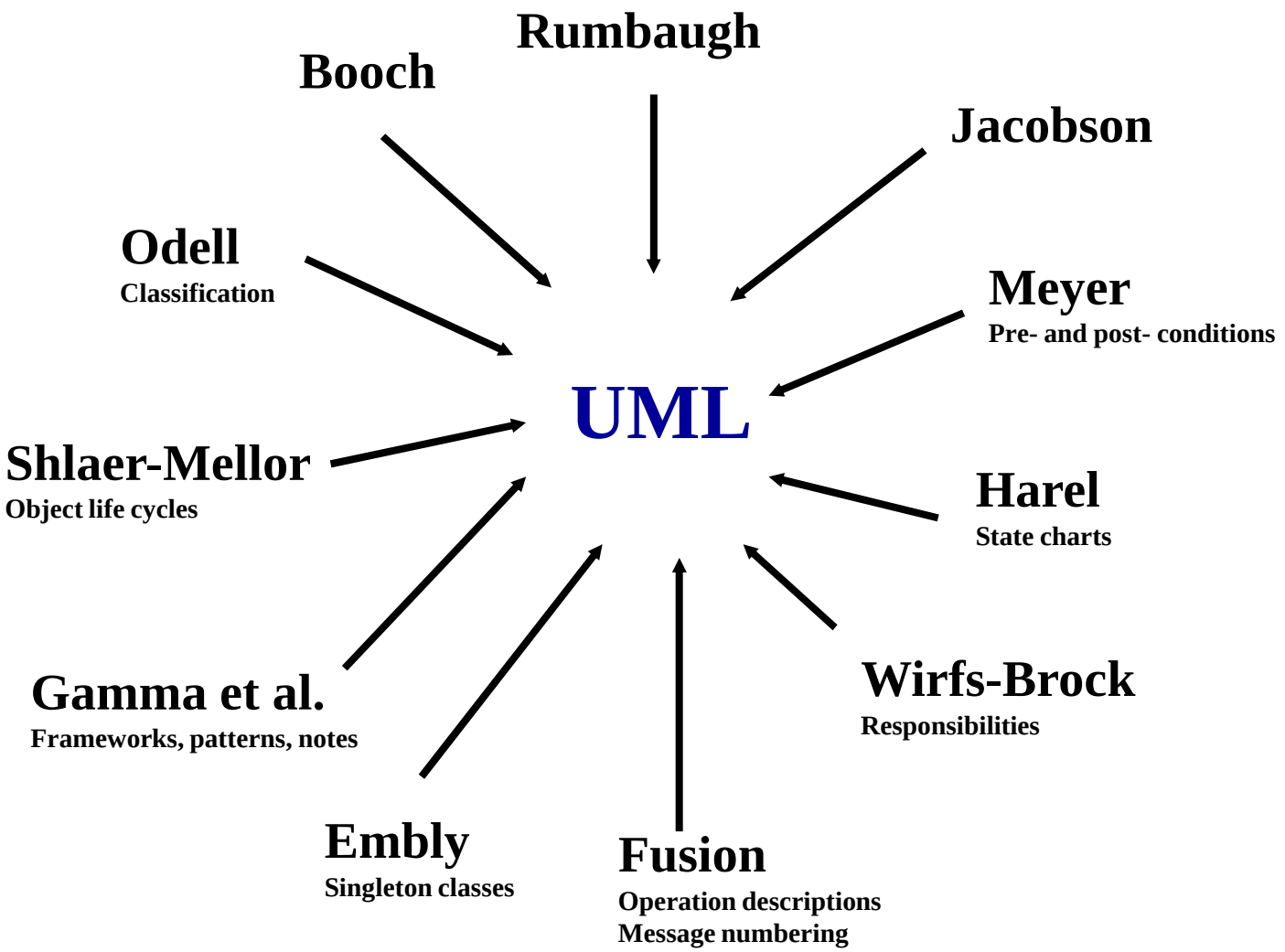
- **A view describes a statistical or a dynamical aspect of the system**
- **For each view**
  - ***Several diagrams are available***
  - ***Example: interactions between objects***
    - **Sequence diagrams**
    - **Collaboration diagrams**
- **Components of views**
  - ***Classes, ports, interfaces, actors, messages, etc.***
- **Mechanisms for extending diagrams**
  - ***Stereotypes, notes, constraints***

I. Introduction to UML

TELECOM PARIS
école nationale supérieure des télécommunications

- Introduction to modeling
- OMG
- UML 1.5 and UML 2.0
- UML for embedded systems and protocols

Origin of UML



The OMG

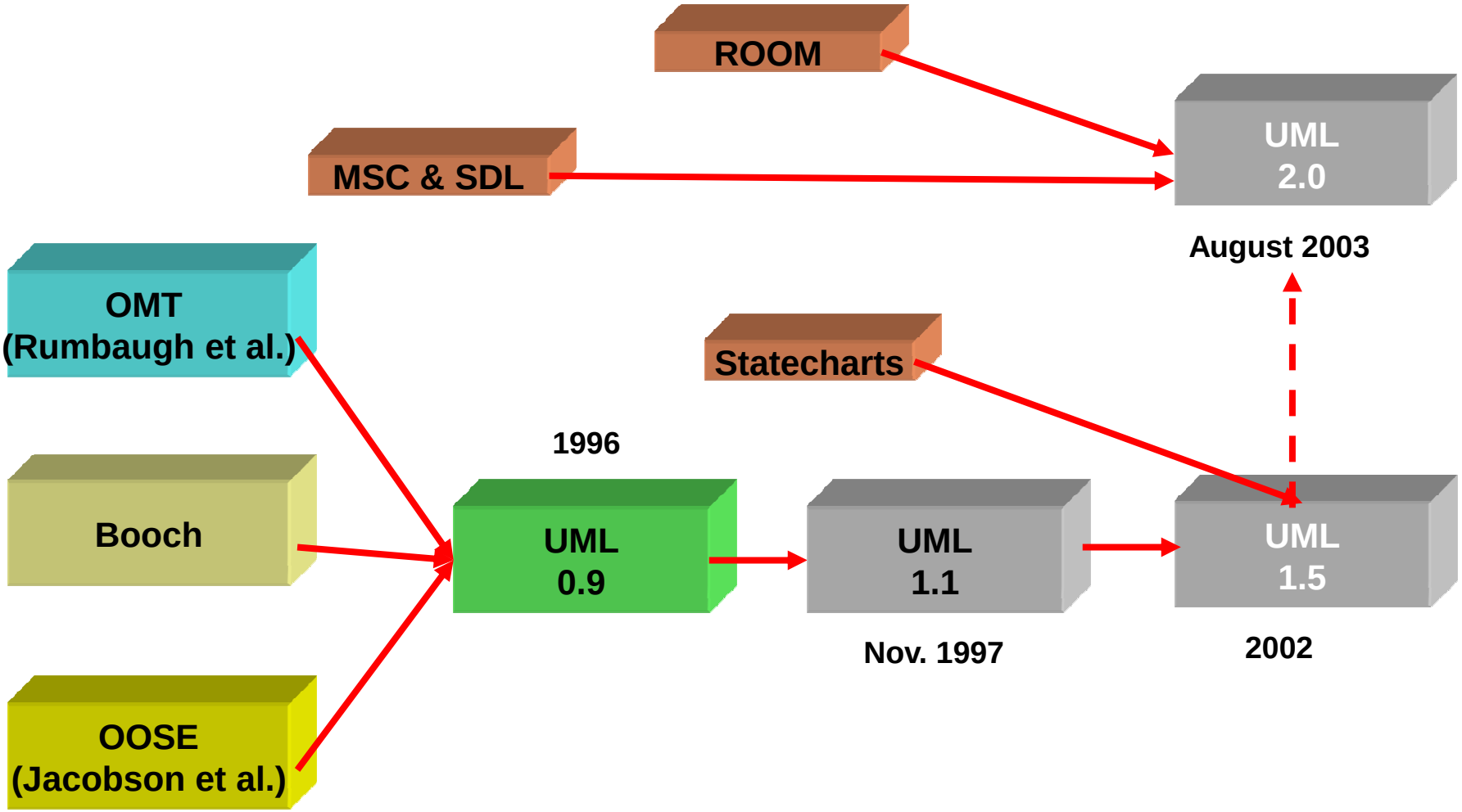
- **Object Management Group**
- **Non-profit organization**
- **Goal: definition of standards related to object-oriented services**
  - ***MOF, UML, XMI, CWM, CORBA (includes IDL, IIOP)***
- **Chronology**
  - ***1989: 11 creating members***
  - ***Nowadays, more than 800 members***
  - ***Members have more or less important vote weight***

I. Introduction to UML

TELECOM PARIS
école nationale supérieure des télécommunications

- Introduction to modeling
- OMG
- UML 1.5 and UML 2.0
- UML for embedded systems and protocols

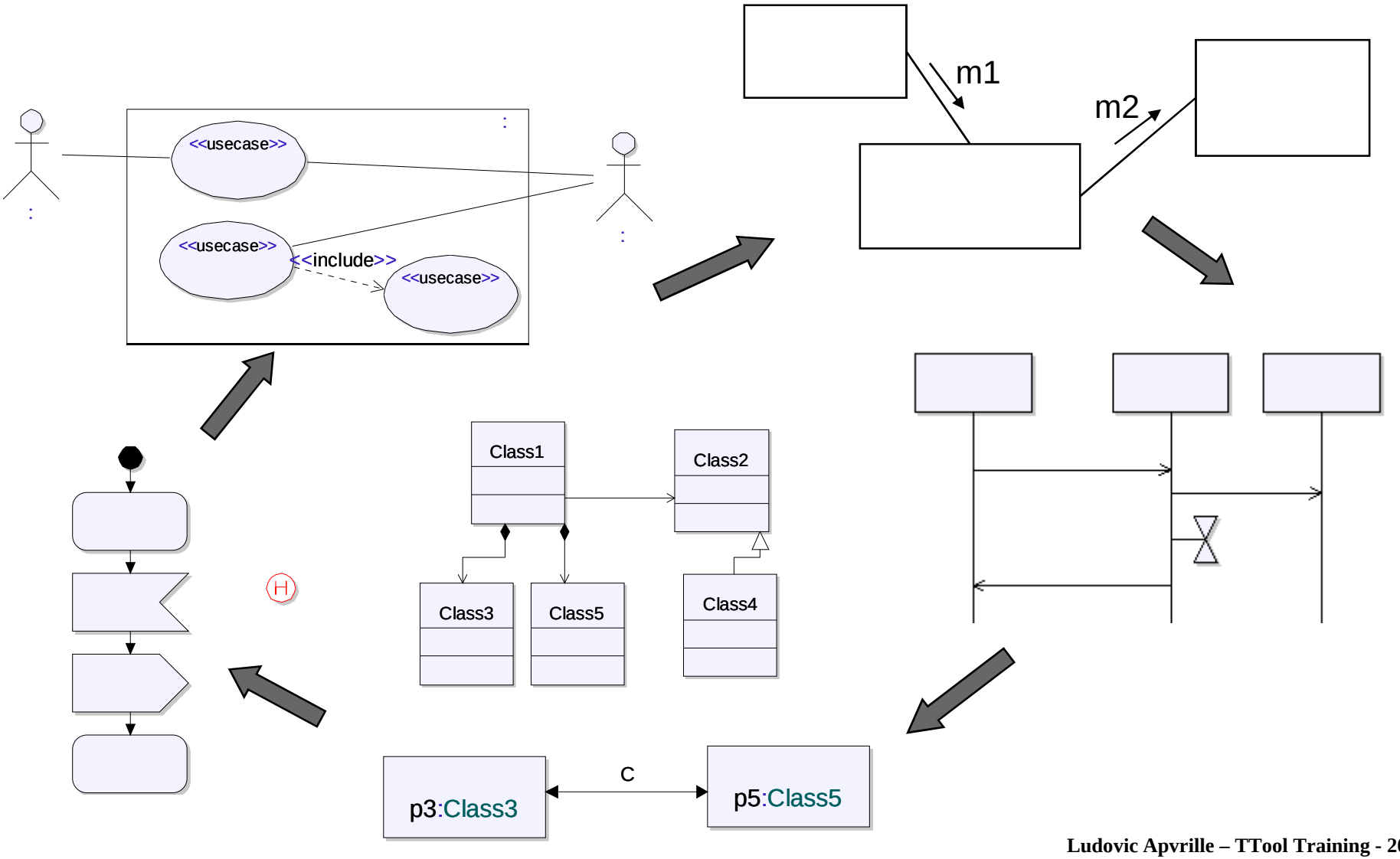
Towards UML 2.0



From UML 1.5 to UML 2.0

1. Use Case diagram		1. Class diagram
2. Class diagram		2. Use case diagram
3. Object diagram		3. Object diagram
4. Statechart diagram	renamed →	4. State machine diagram
5. Activity diagram		5. Activity diagram
6. Sequence diagram		6. Sequence diagram
7. Collaboration diagram	renamed →	7. Communication diagram
8. Component diagram		8. Component diagram
9. Deployment diagram		9. Deployment diagram
	new →	10. Composite structure diagram
	new →	11. Interaction overview diagram
	new →	12. Timing diagram
	new →	13. Package diagram

Overview


<<usecase>>
<<usecase>>
<<include>>
<<usecase>>
m1
m2
Class1
Class2
Class3
Class5
Class4
H
p3:Class3
C
p5:Class5

I. Introduction

- Introduction to modeling
- OMG
- UML 1.5 and UML 2.0
- UML for embedded systems and protocols

UML for Embedded Systems

TELECOM PARIS
école nationale supérieure des télécommunications

- **Specificity of embedded systems and protocols**
  - ***Strict constraints***
    - **Performance constraints, real-time constraints, etc.**
    - **Critical aspect**
  - ***Limited resources***
  - ***Interactions between components***
- **Specific UML methodology**
  - ***Make use of some UML diagrams rather than others***
  - ***Make use of simulation techniques as soon as possible in the development cycle***
    - **Critical systems**
- **Specific UML toolkits**
  - ***Profiles***

A UML Methodology Focused on Embedded Systems and Protocols

TELECOM PARIS
école nationale supérieure des télécommunications

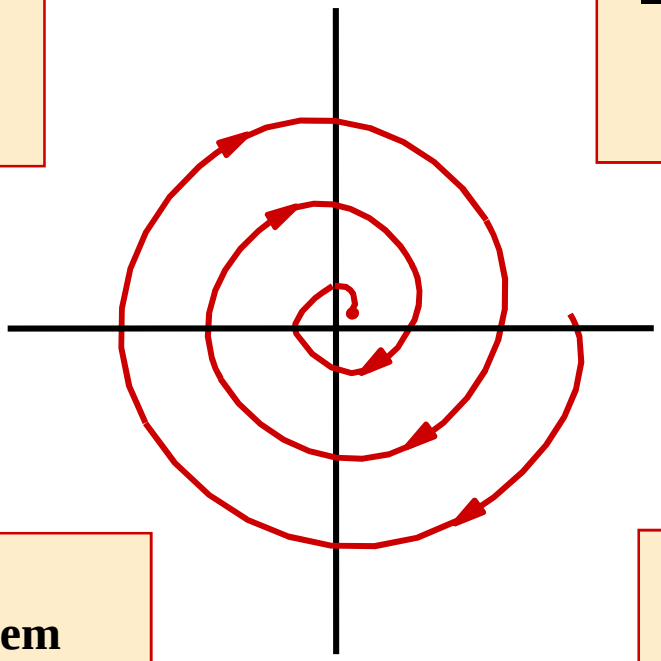
Detailed design

Behavior of the system

Validation of the system

Simulation

Code generation



Design

Classes of the system

Architecture of the system

Analysis

Use case

First class diagram

Relevant scenarios

Analysis Stage

TELECOM PARIS
école nationale supérieure des télécommunications

- **Purpose**
  - *Analysis of the requirements of the system*
- **Steps**
  - *Identification of use cases of the systems, and structuring of these cases*
    - **Use case diagram**
  - *First scenarios emphasizing exchanges between actors of the system and the system itself*
    - **Sequence diagram**
  - *Identification of the main classes of the system*
    - **Class diagram**
  - *Refined Scenarios*
    - **Classes identified at previous step are introduced into previously performed scenarios**
    - **Sequence diagrams**

Design stage

TELECOM PARIS
école nationale supérieure des télécommunications

- **Purpose**
  - ***Structure the system under the form of classes and relations among those classes***
- **Steps**
  - ***Identification of secondary classes***
    - **Class diagram**
  - ***Identification of relations between classes***
    - **Association, aggregation, specialization, etc.**
    - **Class diagram**
  - ***Modeling of class hierarchy***
    - **Class diagram / Composite structure diagram**
  - ***Modeling message exchange between classes***
    - **Class diagram / Composite structure diagram**

Detailed Design Stage

TELECOM PARIS
école nationale supérieure des télécommunications

- **Purpose**
  - ***Describe the behavior of the system***
- **Steps**
  - ***Description of classes behavior***
    - **Signals, operations**
    - **State machine diagram / Activity diagram**
  - ***Description of the system dynamics***
    - **Creation / destroy of instances**
    - **State machine diagram / Activity diagram**
  - ***Refinement of relations among classes***
    - **Specialization**
    - **Class diagram**

Validation Stage

TELECOM PARIS
école nationale supérieure des télécommunications

Purpose

- ***Check that the behavior of the system corresponds to the targeted one***

Steps

- ***Simulation***
  - **Validation of the modeling as soon as possible**
  - **Modeling made at analysis stage and design stages are compared**
  - **Use case diagrams, Sequence diagrams vs. class diagrams, composite structure diagrams and state machine diagrams (or activity diagrams)**
- ***Implementation***
  - **Automatic code generation**
    - ***C, C++, Java, Ada***
  - **Class diagrams, composite structure diagrams and state machine diagrams (or activity diagrams)**

UML Diagrams for Embedded Systems and Protocols

TELECOM PARIS
école nationale supérieure des télécommunications

1. Class diagram
2. Use Case diagram
3. Object diagram
4. State machine diagram
5. Activity diagram
6. Sequence diagram
7. Communication diagram
8. Component diagram
9. Deployment diagram
10. Composite structure diagram
11. Interaction overview diagram
12. Timing diagram
13. Package diagram

UML Toolkits for Embedded Systems

TELECOM PARIS
école nationale supérieure des télécommunications

- **Goal: edition of diagrams, animation, code generation**
- **TAU G2**
  - *www.telelogic.com*
- **Rhapsody**
  - *www.ilogix.com*
- **ARTiSAN Real-Time Studio**
  - *www.artisansw.com*
- **(ROSE-RT)**
  - *www.rational.com*
- **TTool!**
  - ***www.eurecom.fr/~apvrille/TURTLE***

Books

- Michael Jesse Chonoles, James A. Schardt, “UML 2 for Dummies”, Wiley, 2003, ISBN 0-7645-2614-6
- Laurent Doldi, “UML 2 Illustrated - Developing Real-Time & Communications Systems”, TMSO , 2003, ISBN 2-9516600-1-4
  http://perso.wanadoo.fr/doldi/sdl/umlbook.htm
- Tom Pender, “UML Bible”, John Wiley & Sons, 2003, ISBN 0764526049
- Luciano Lavagno et al., “UML for Real: Design of Embedded Real-Time Systems”, Kluwer Academic Publishers, ISBN 1-4020-7501-4

Structured vs. Object-Oriented Programming

TELECOM PARIS
école nationale supérieure des télécommunications

- **Structured programming**
  - *= Imperative programming*
  - *Programs are structured into subprograms to manage complexity*
  - *Emphasizes functions*

- **But: data is what is most likely to be stable in the life of a program**

- **Object-oriented programming**
  - *First focused on data rather than on functions*
  - *A computer program is composed of a collection of individual units, or objects*
  - *Challenge of programmers: distribute responsibility over objects*

Fundamentals of Objects

TELECOM PARIS
école nationale supérieure des télécommunications

- **Object-oriented paradigm**
  - ***A problem is addressed with object-oriented concepts***
- **An object is an abstraction of data themselves containing abstractions of functions**
- **Objects exchange messages**
  - ***They collaborate together to achieve predefined tasks***

Definitions

- **Booch, *Object-Oriented Design with Applications***
  - ***An object is made of a state, a behavior and of an identity***
  - ***The terms "instance" and "object" are interchangeable***
- **Rumbaugh et al., *Object-oriented Modeling and Design***
  - ***Design, abstraction or "thing" whose frontiers and significations are very close to the addressed problem***
- **Jacobson, *Object-oriented Software Engineering, a Use-Case Driven Approach***
  - ***An object is en entity able to save a state (information) and that offers a given number of operations to consult this state or to modify it***

Examples of Objects

Ludovic
Ludovic's last name Ludovic's address brut Income net Income taxable income

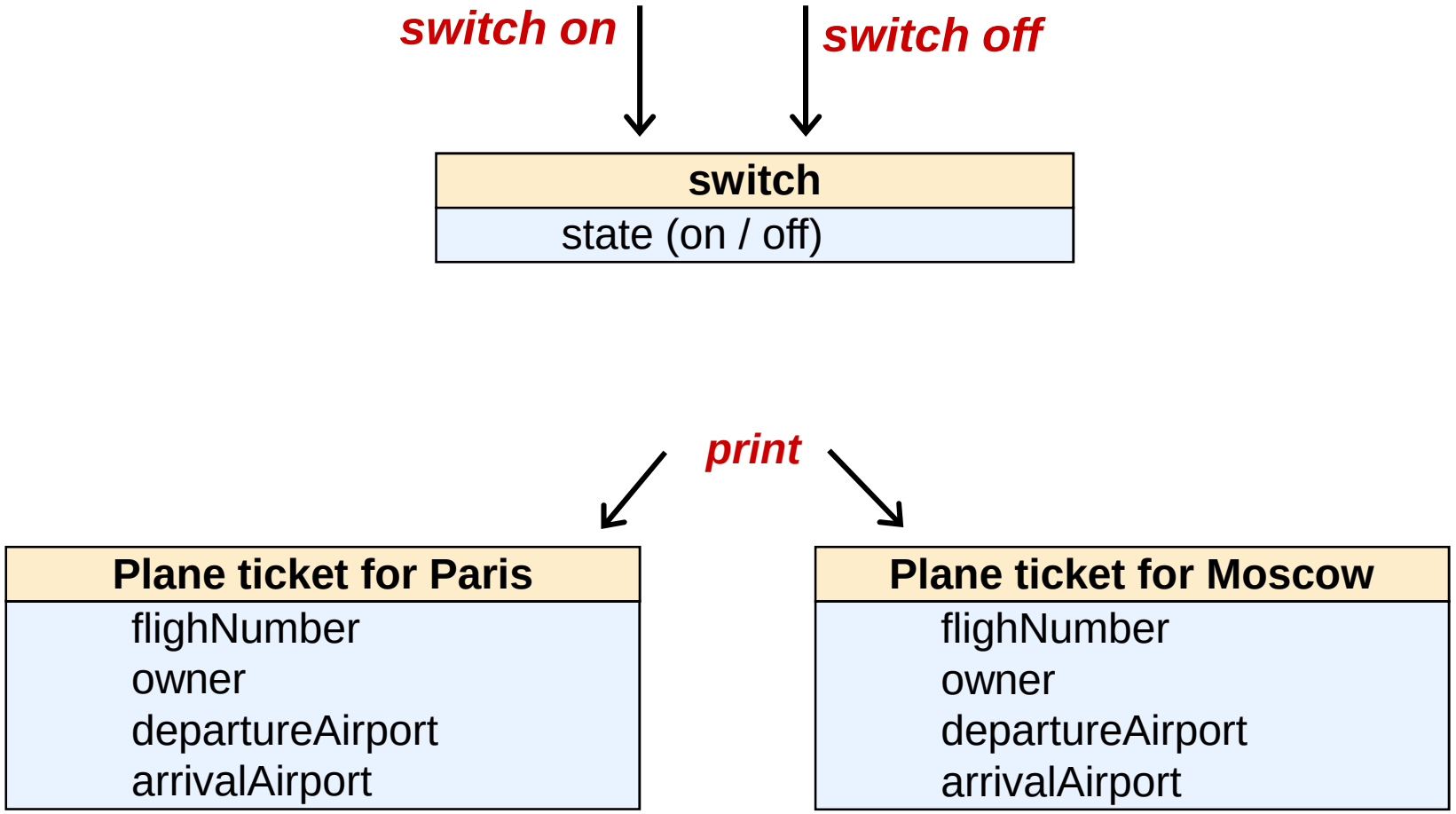
Course on UML
number title author duration

switch
state (on / off)

Plane ticket for Paris
fligh Number owner departureAirport arrivalAirport

Plane ticket for Moscow
fligh Number owner departureAirport arrivalAirport

Example of Objects and Messages



States Characterized by Attributes

TELECOM PARIS
école nationale supérieure des télécommunications

- **The value of the attributes defines the state of the object**
  - ***Static characteristics***
  - ***Dynamic characteristics***
- **Exercises**
  - ***What are the attributes of a checking account?***
- **Issue of the access rights to attributes!**

A Behavior Based on Operations

- **Operations are all possible actions on an object**
  - ***A response from the object might be required***
- **An object O1 can communicate with an object O2**
  - ***Invocation of an operation of O2***

- **Issue of the access rights to operations!**

Notion of Message

- One-way communication between two objects
- Flow of control with information passed from the sender to the receiver.
- May have parameters that convey values
- Can be
  - *a signal*
  - *a call*
  - *a return*
  - *create*
  - *destroy*

Classes and Objects

TELECOM PARIS
école nationale supérieure des télécommunications

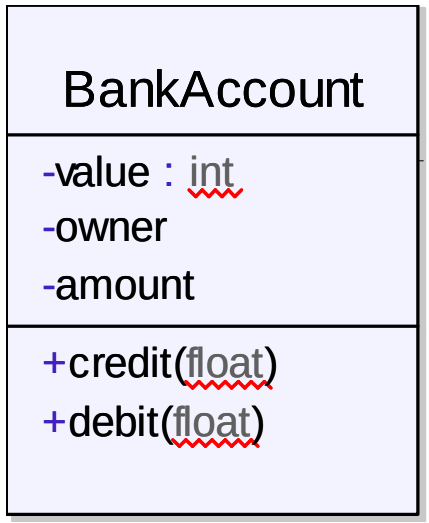
- **Same duality as “type” and “variable”**
- **Class**
  - ***Unit that eases the definition of objects sharing common characteristics***
    - **Attributes, operations**
- **Object**
  - ***An entity of the real world built upon an abstract unit***
  - ***Instance of a class***
    - **Attributes and operations are defined in the corresponding class**
  - ***State of an object = value of its attribute at a given time***
  - ***Behavior of an object = set of operations it can perform when reacting from messages coming from other objects***

Example of Classes in UML

Name of the class

List of attributes
- : private

Liste of operations
+ : public



UML Comment

A basic bank account has no granted overdraft
or maximum deposit amount

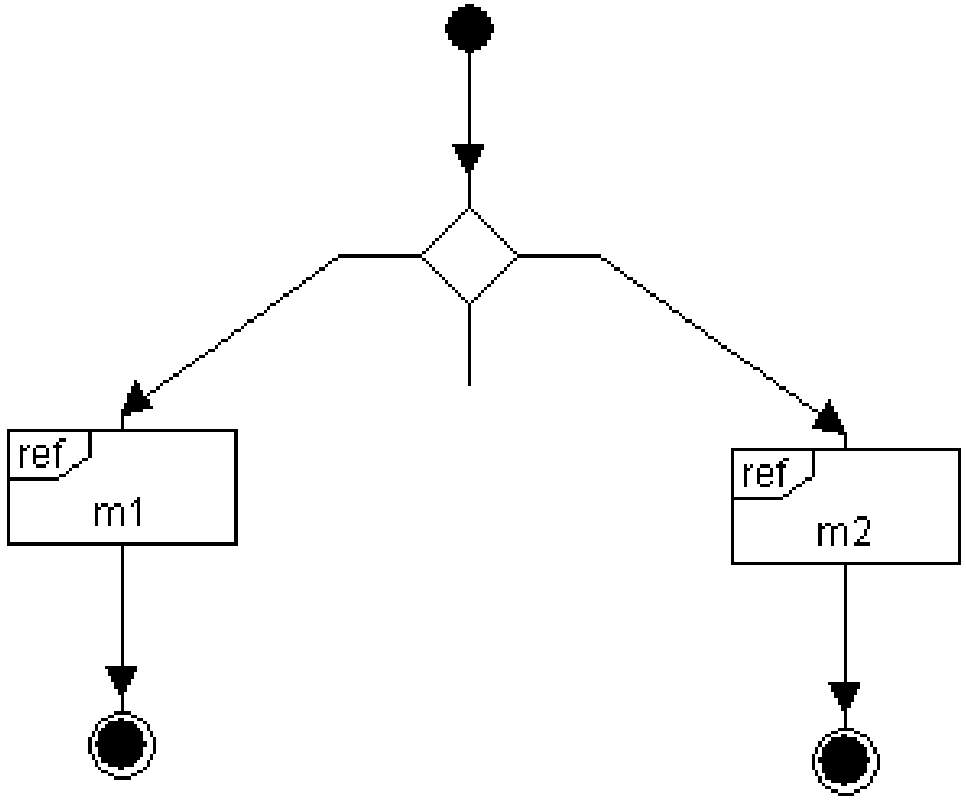
Attributes and Operations of Instances

- **All instances of the same class have the same behavior**
  - ***Operations***
- **Each instance has its own state**
  - ***Attributes of the instance***
    - **Their value may be different for each instance**
- **Are global variables possible in the object-oriented paradigm?**
  - ***Attributes common to all instances of a class***

Fundamentals of Object-Oriented Paradigm

- **Modularity**
  - ***The computer program is built entirely inside classes***
- **Encapsulation**
  - ***Information hiding***
    - **No need to have a knowledge of the inside of a class to use it -> only the knowledge of its interface is required**
- **Abstraction**
  - ***Objects: abstraction of the real world***
  - ***Classes: abstraction of objects***
- **Reusability**

Interaction Overview Diagrams

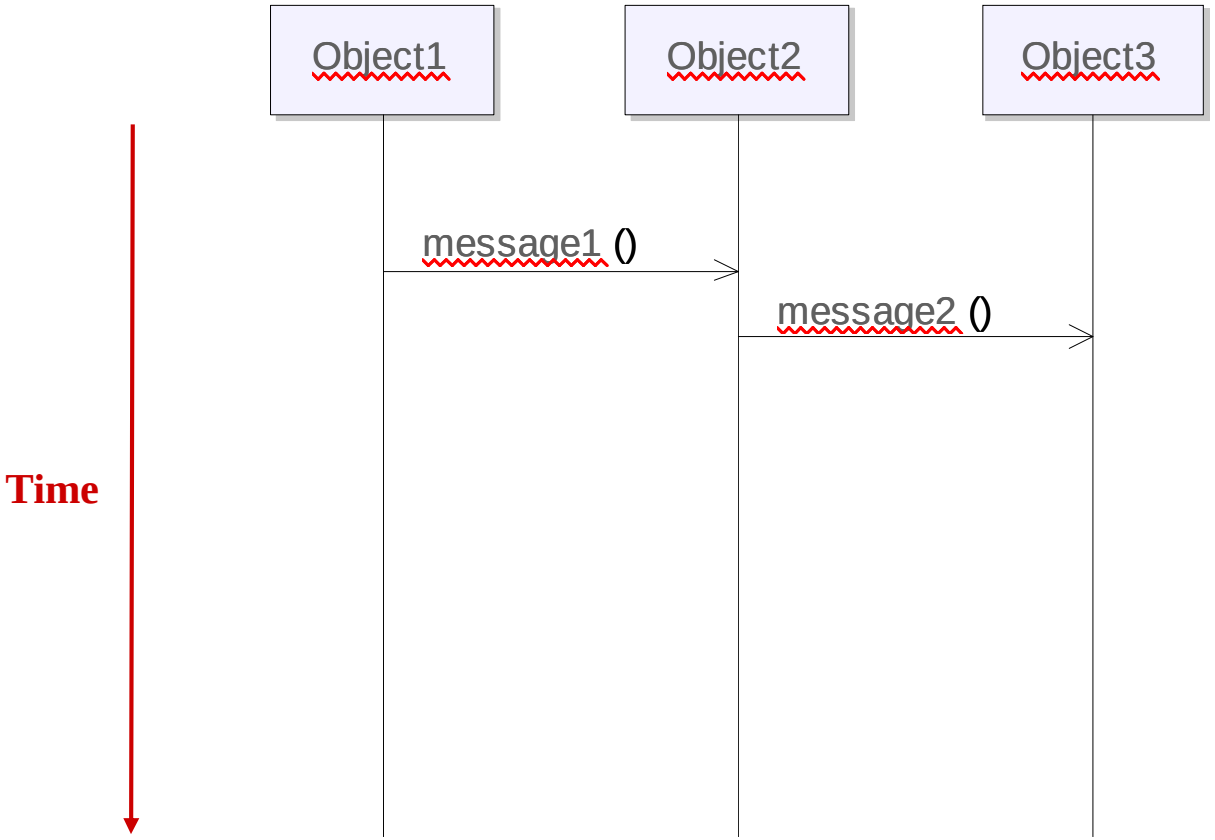


UML Sequence Diagrams

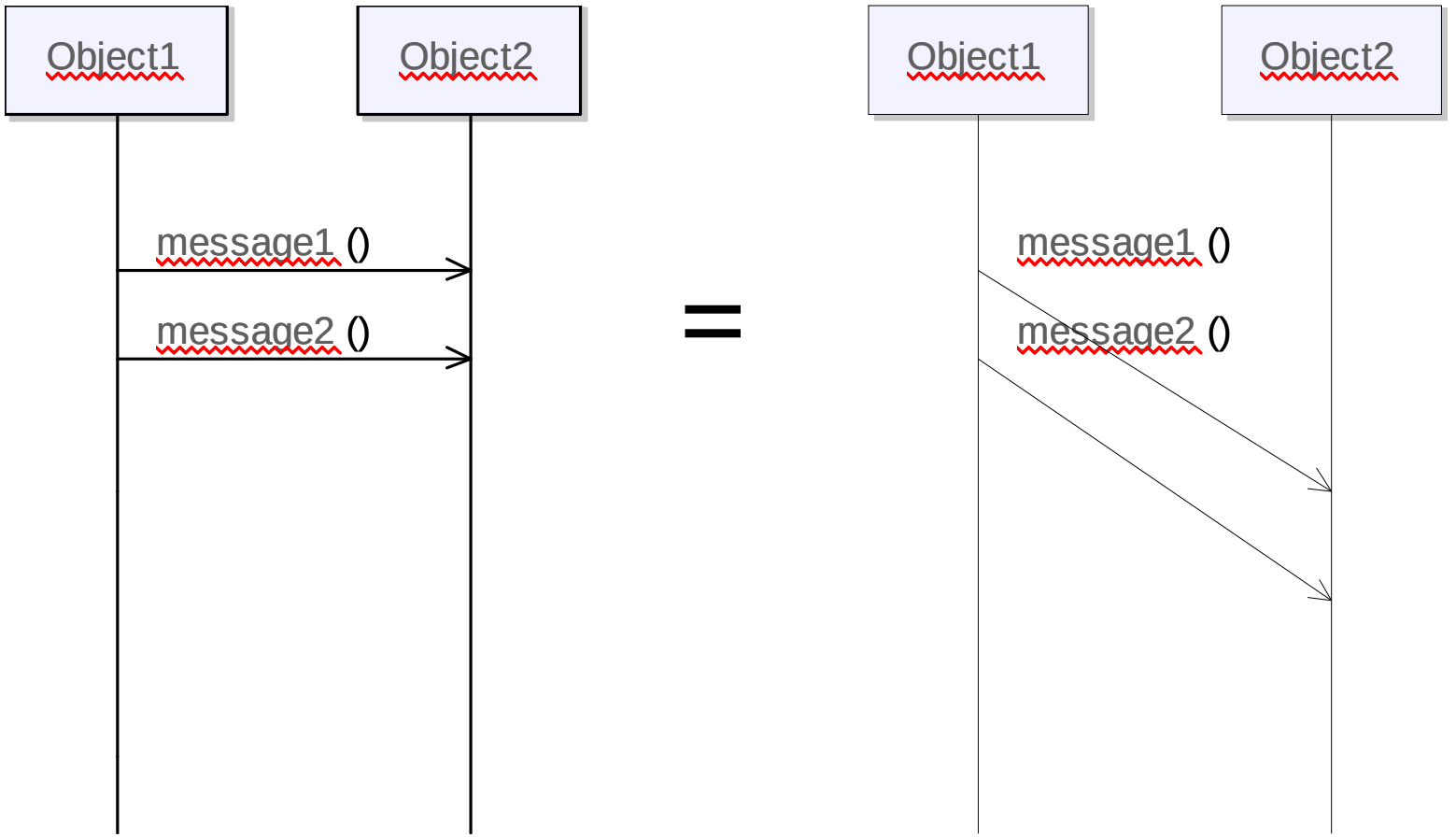
TELECOM PARIS
école nationale supérieure des télécommunications

- **Basics of sequence diagrams**
  - *Gives clear visual clues to possible flows of control over time*
  - *Emphasizes time ordering*
  - *Shows object lifeline*
  - *Shows the focus of control*
- **UML 1.5**
  - *Notion of message (or stimulus) and of lifeline*
  - *Observation of time*
  - *Temporal constraints*
  - *Activation of an object*
- **UML 2.0**
  - *Suspension, interaction, duration constraints*

Basic Syntax

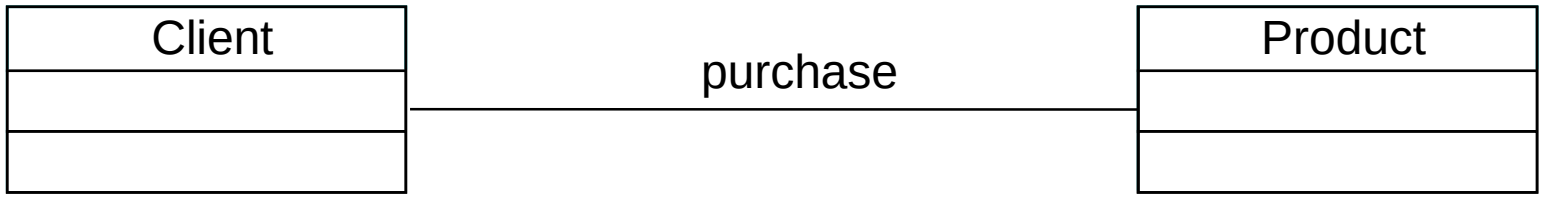


Basic Syntax (Cont.)

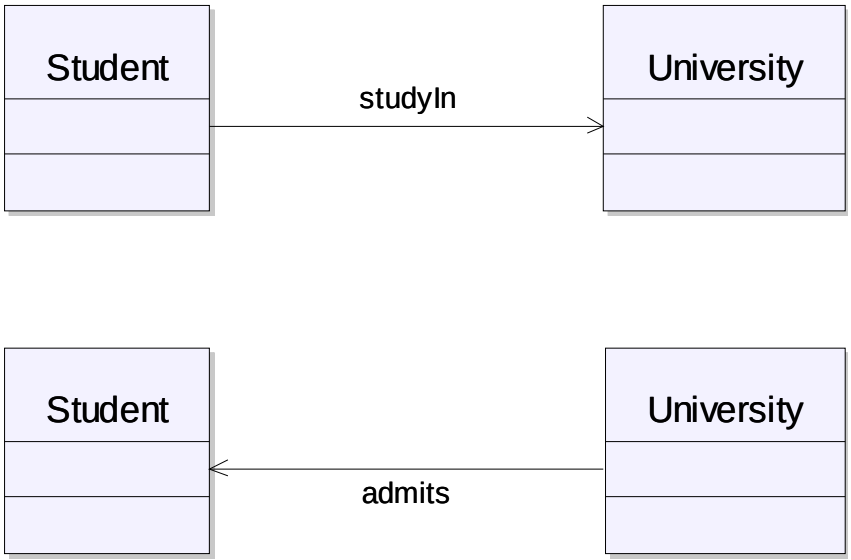


Association

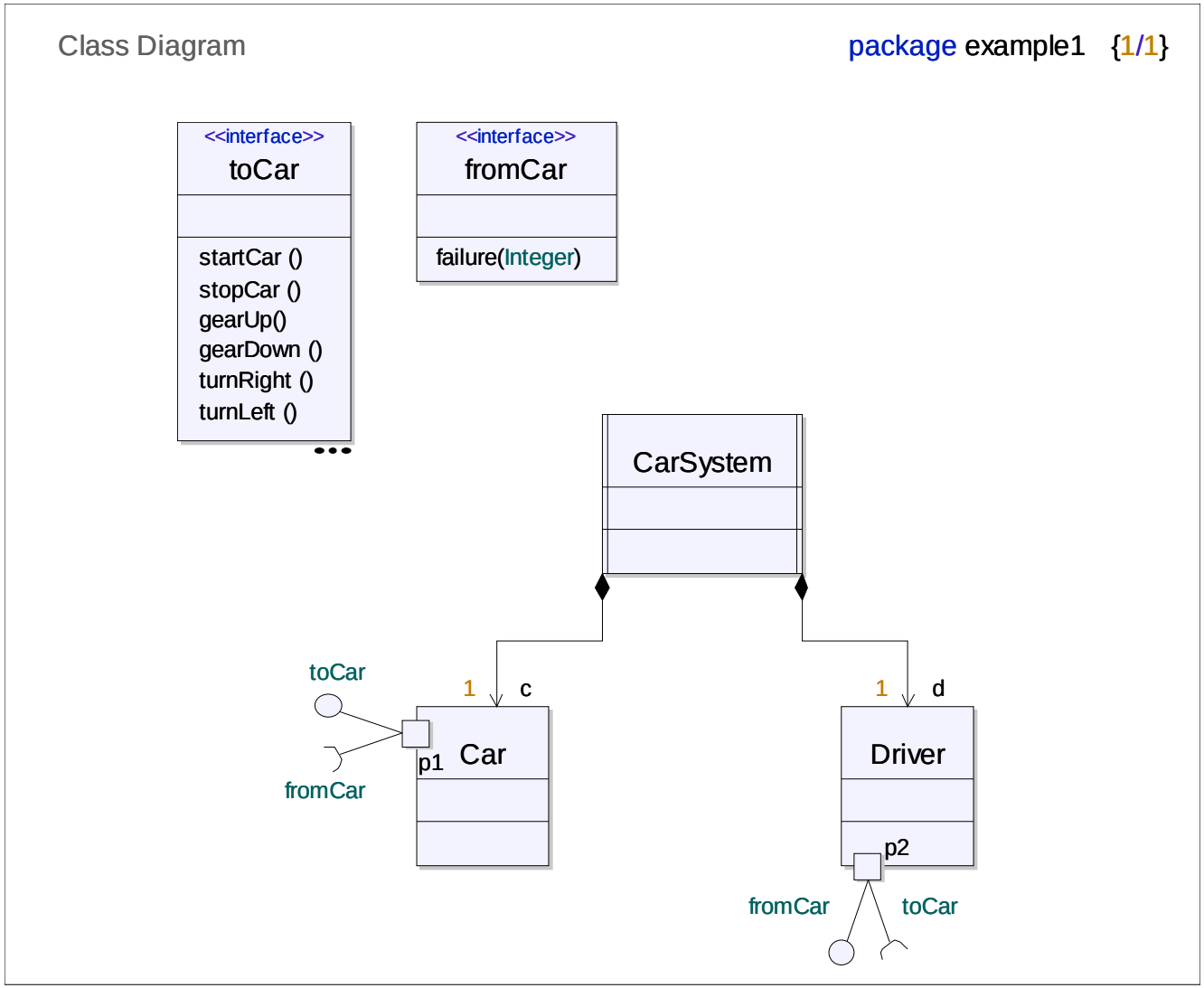
Association name



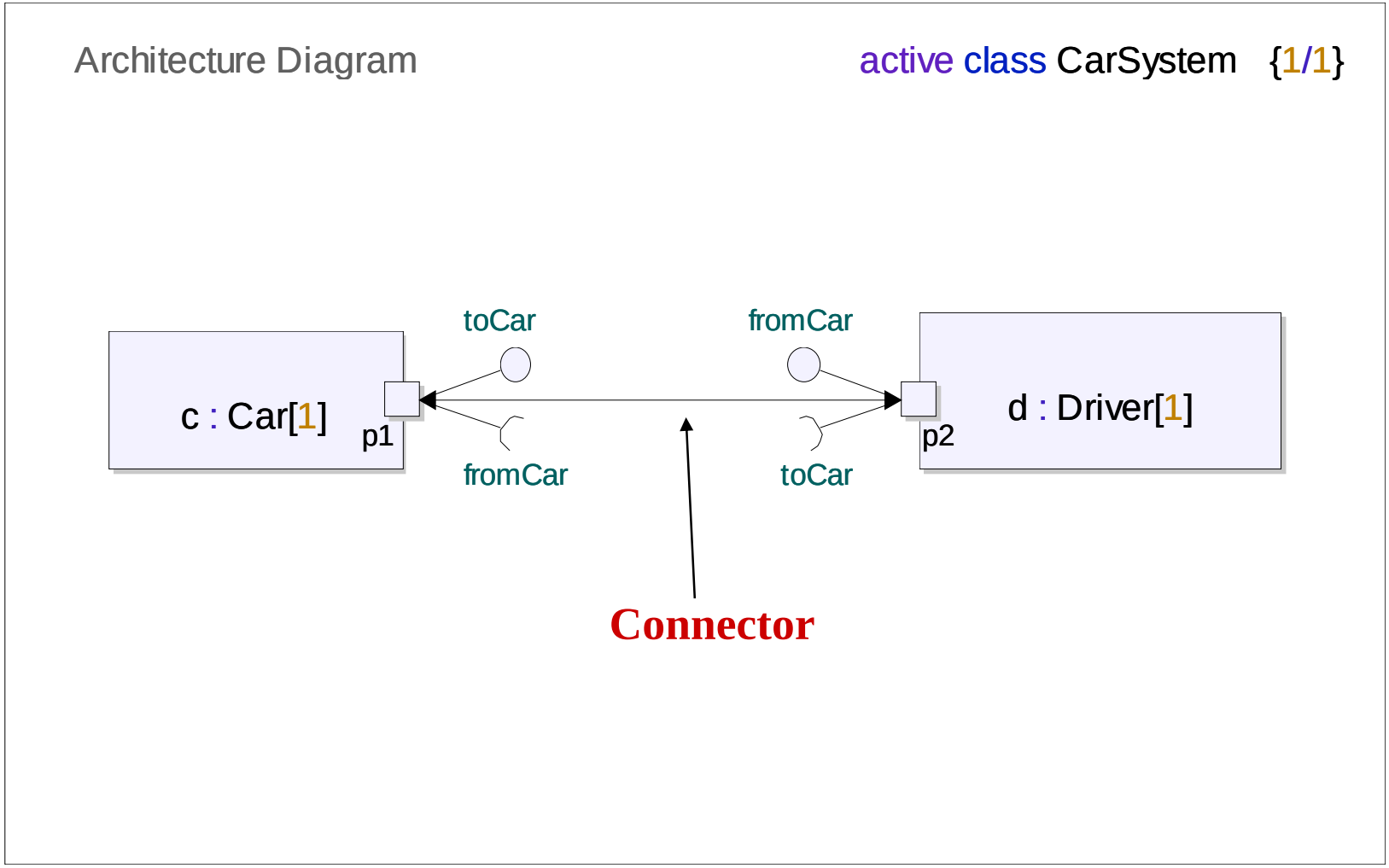
Association with navigability



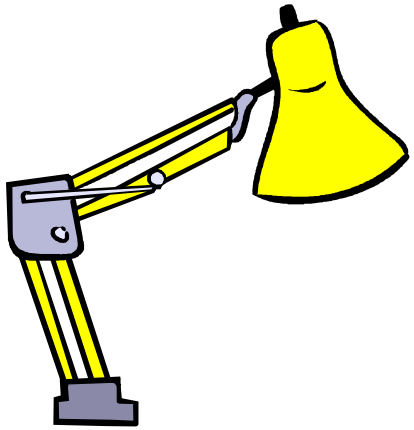
Example: At Class Diagram Level

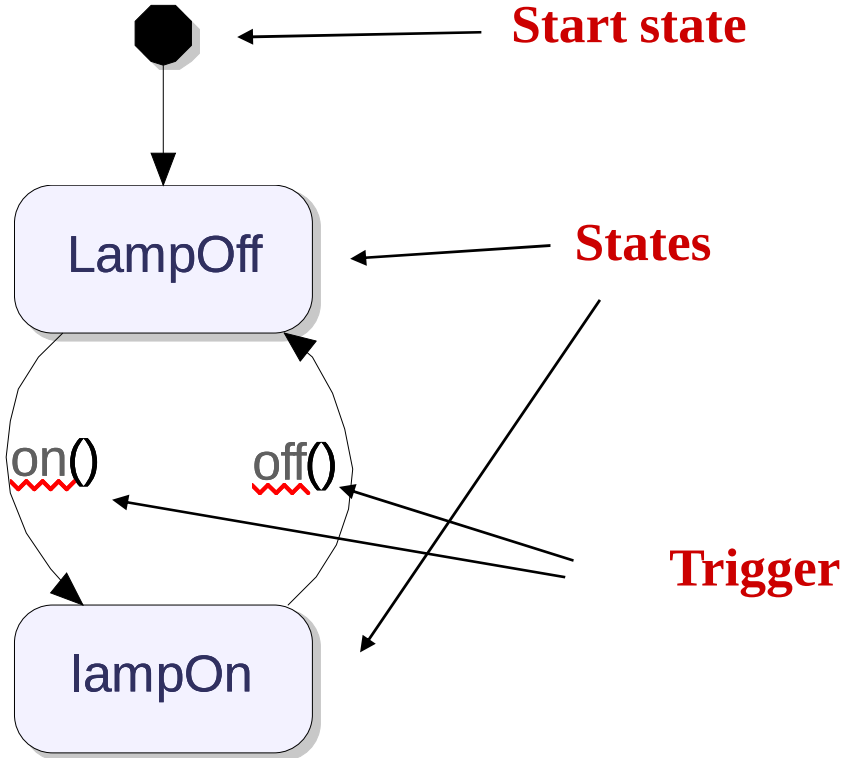


Example: At Composite Structure Diagram Level



Example of State Machine Diagrams





Activity Diagrams