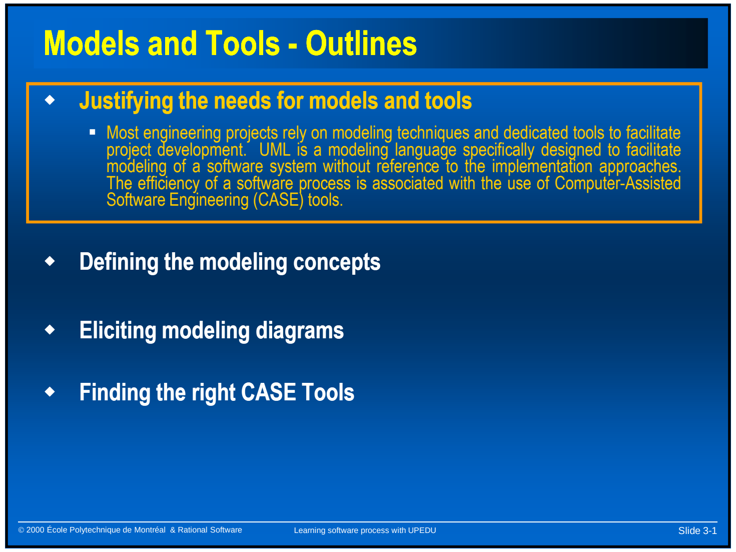
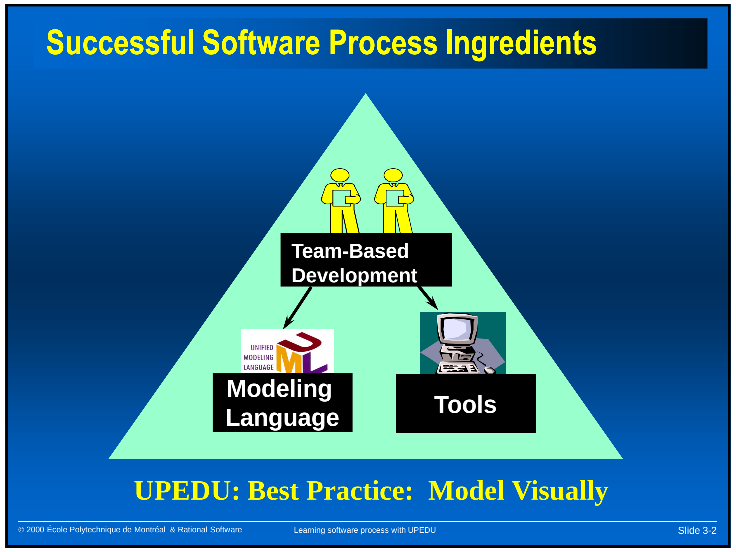


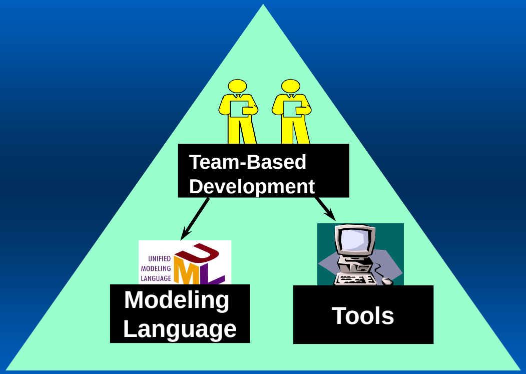
Models and Tools - Outlines

- **Justifying the needs for models and tools**
  - Most engineering projects rely on modeling techniques and dedicated tools to facilitate project development. UML is a modeling language specifically designed to facilitate modeling of a software system without reference to the implementation approaches. The efficiency of a software process is associated with the use of Computer-Assisted Software Engineering (CASE) tools.
- **Defining the modeling concepts**
- **Eliciting modeling diagrams**
- **Finding the right CASE Tools**

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-1

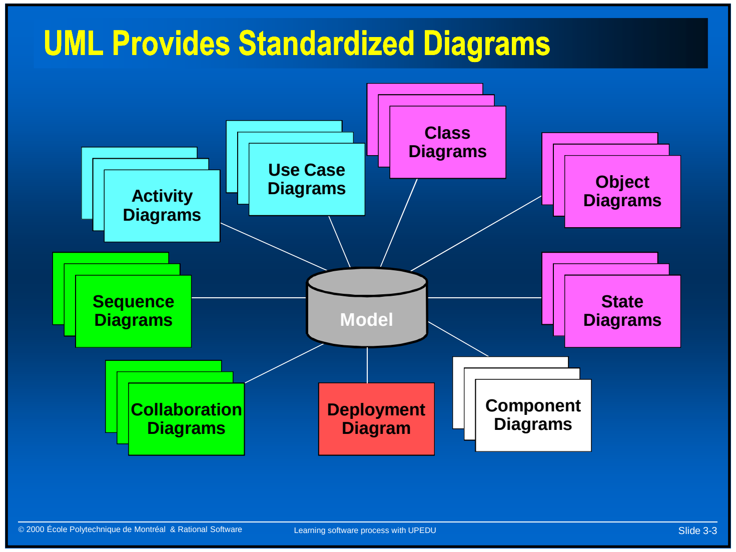


Successful Software Process Ingredients

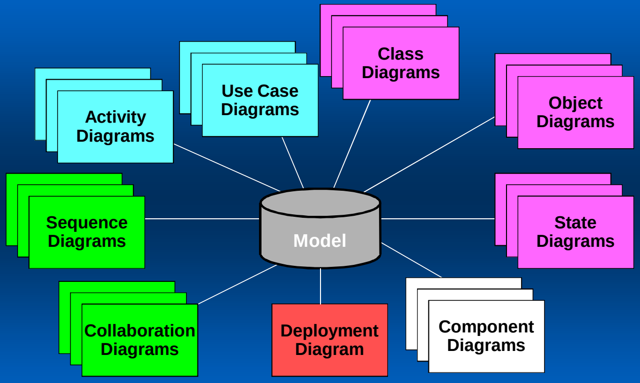


UPEDU: Best Practice: Model Visually

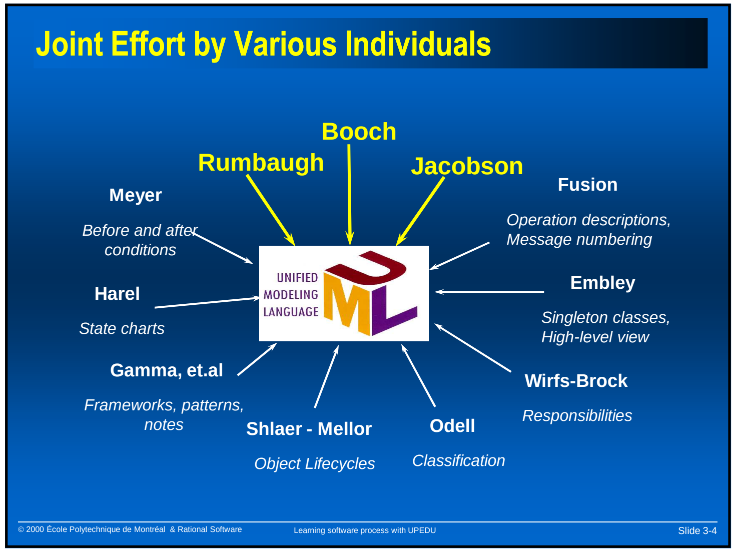
© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-2



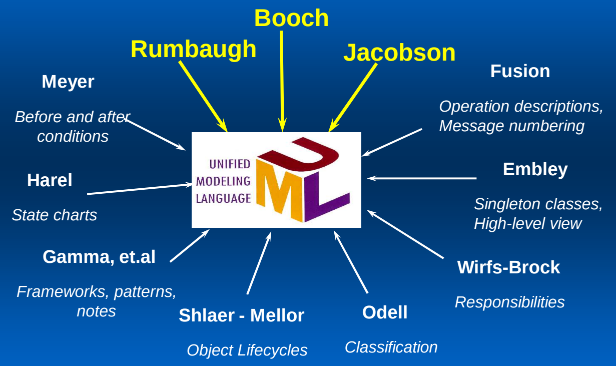
UML Provides Standardized Diagrams



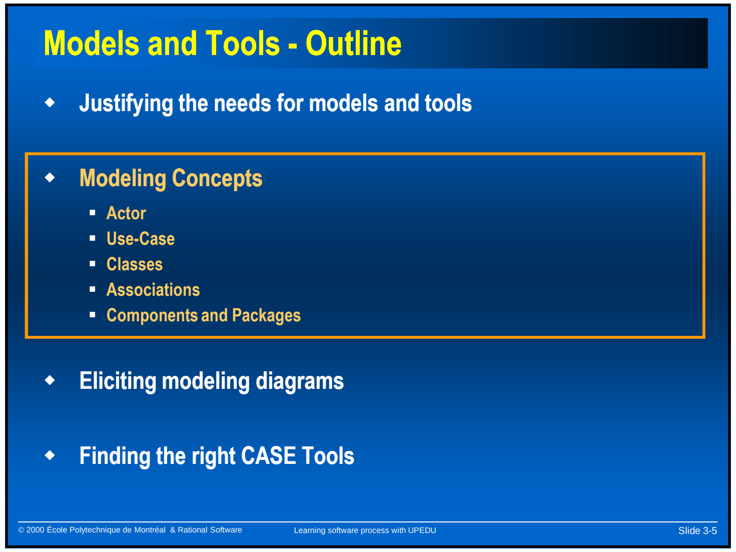
© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-3



Joint Effort by Various Individuals


Booch
Rumbaugh
Jacobson
Meyer
Before and after conditions
Fusion
Operation descriptions, Message numbering
Harel
State charts
Embley
Singleton classes, High-level view
UNIFIED MODELING LANGUAGE
Gamma, et.al
Frameworks, patterns, notes
Shlaer - Mellor
Object Lifecycles
Odell
Classification
Wirfs-Brock
Responsibilities

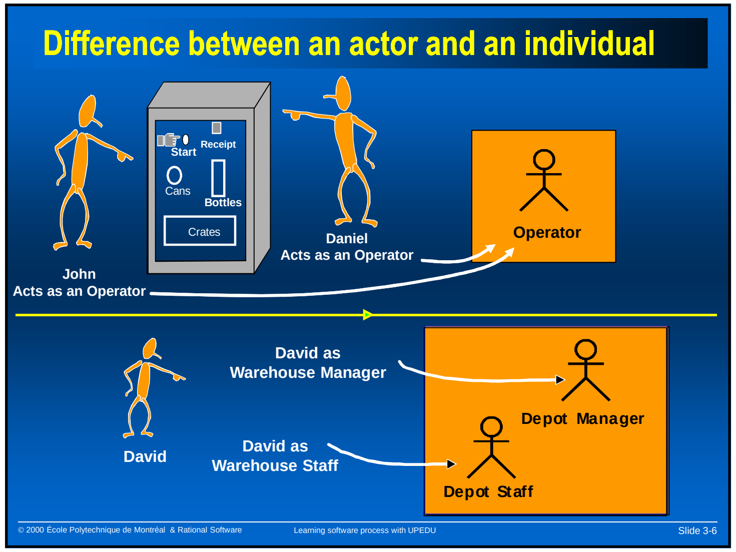
© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-4



Models and Tools - Outline

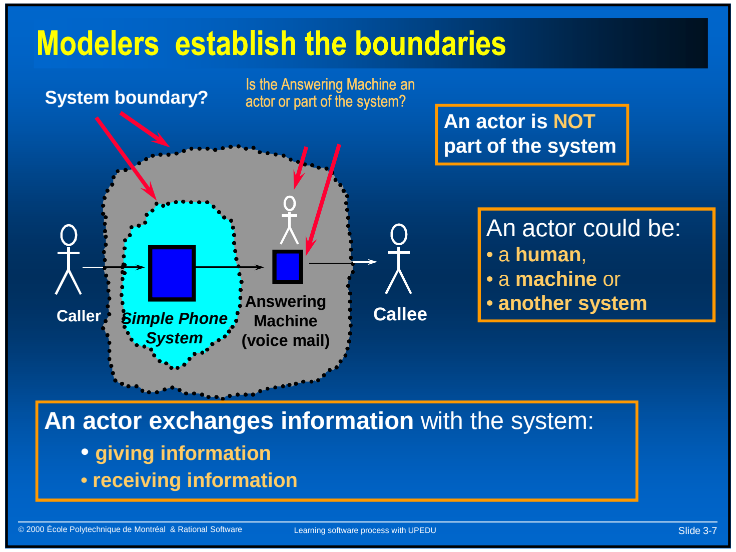
- Justifying the needs for models and tools
- Modeling Concepts
  - Actor
  - Use-Case
  - Classes
  - Associations
  - Components and Packages
- Eliciting modeling diagrams
- Finding the right CASE Tools

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-5



Difference between an actor and an individual

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-6



Modelers establish the boundaries

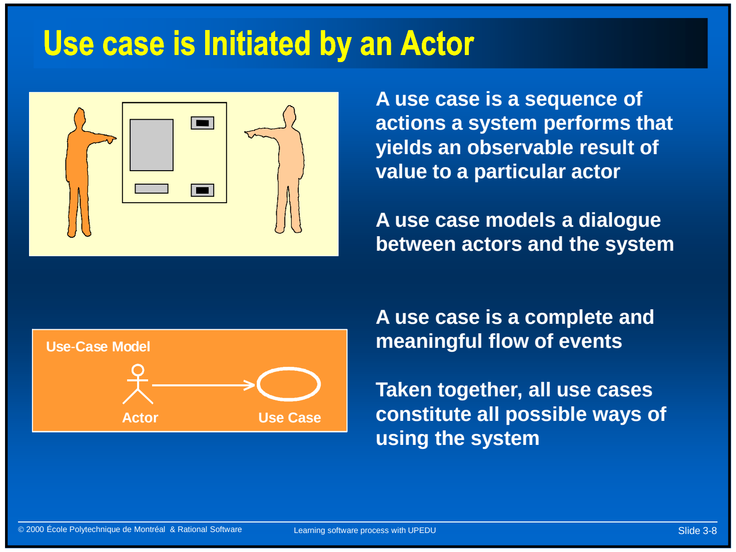
System boundary?
Is the Answering Machine an actor or part of the system?
Caller
Simple Phone System
Answering Machine (voice mail)
Callee

An actor is NOT part of the system

An actor could be:
- a human,
- a machine or
- another system

An actor exchanges information with the system:
- giving information
- receiving information

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-7



Use case is Initiated by an Actor

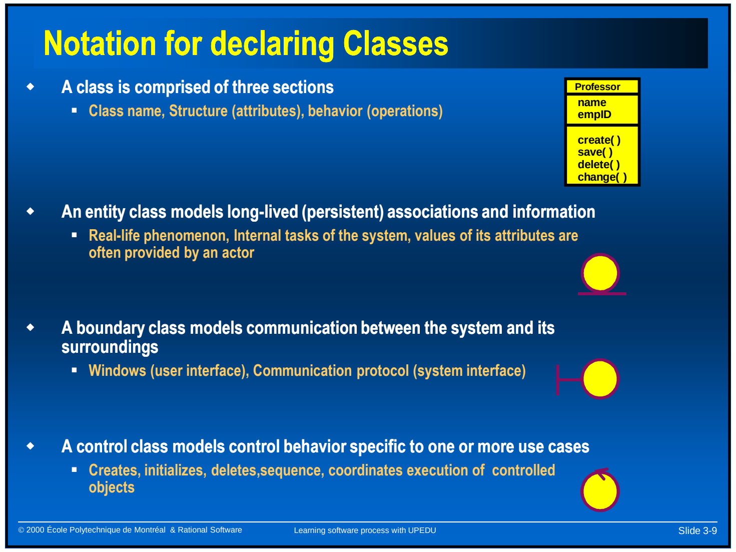
A use case is a sequence of actions a system performs that yields an observable result of value to a particular actor

A use case models a dialogue between actors and the system

A use case is a complete and meaningful flow of events

Taken together, all use cases constitute all possible ways of using the system

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-8



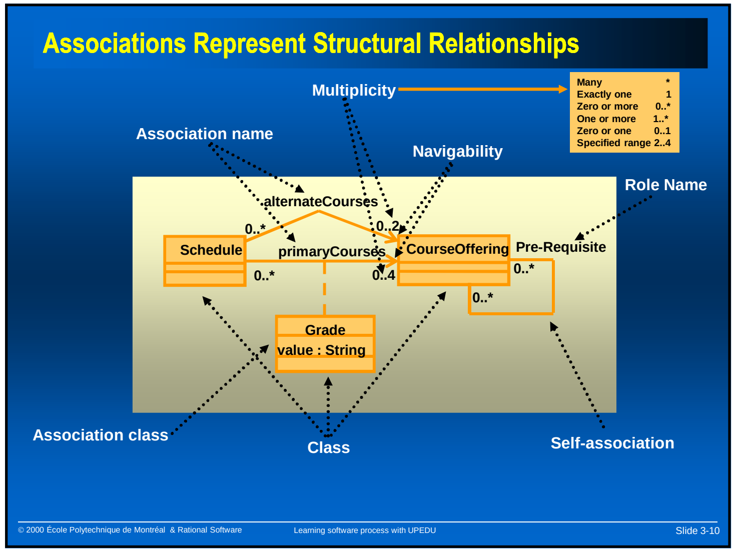
Notation for declaring Classes

- A class is comprised of three sections
  - Class name, Structure (attributes), behavior (operations)

Professor
name
empID
create( )
save( )
delete( )
change( )

- An entity class models long-lived (persistent) associations and information
  - Real-life phenomenon, Internal tasks of the system, values of its attributes are often provided by an actor
- A boundary class models communication between the system and its surroundings
  - Windows (user interface), Communication protocol (system interface)
- A control class models control behavior specific to one or more use cases
  - Creates, initializes, deletes,sequence, coordinates execution of controlled objects

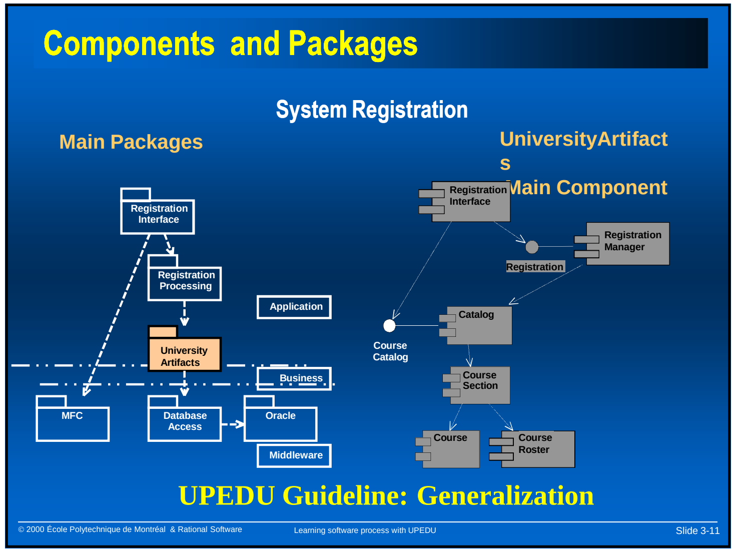
© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-9



Associations Represent Structural Relationships

Many	*
Exactly one	1
Zero or more	0..*
One or more	1..*
Zero or one	0..1
Specified range	2..4

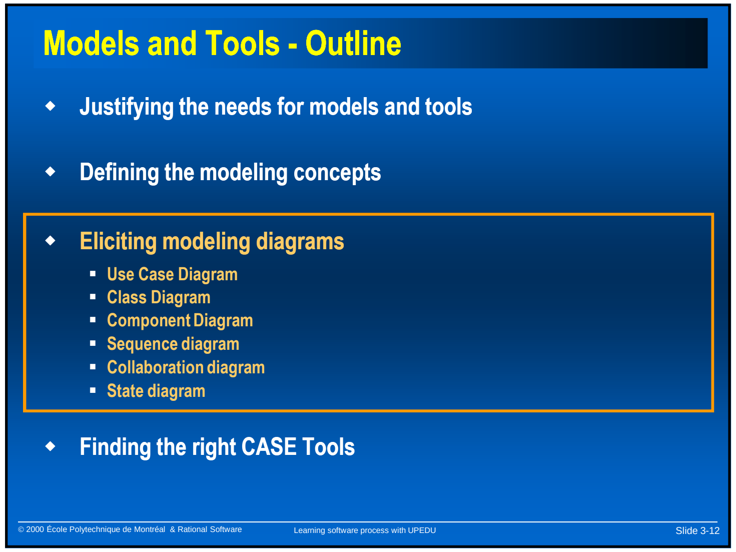
© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-10



Components and Packages

UPEDU Guideline: Generalization

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-11

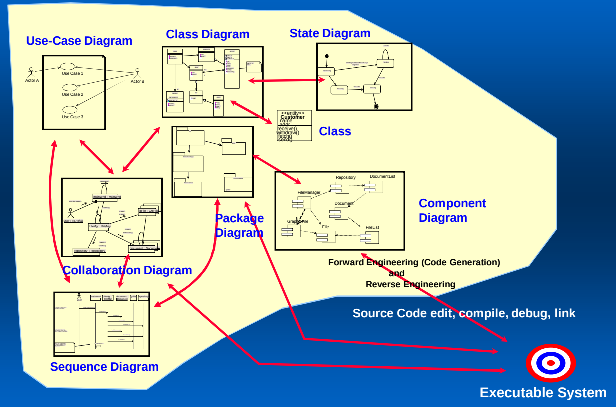


Models and Tools - Outline

- Justifying the needs for models and tools
- Defining the modeling concepts
- Eliciting modeling diagrams
  - Use Case Diagram
  - Class Diagram
  - Component Diagram
  - Sequence diagram
  - Collaboration diagram
  - State diagram
- Finding the right CASE Tools

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-12

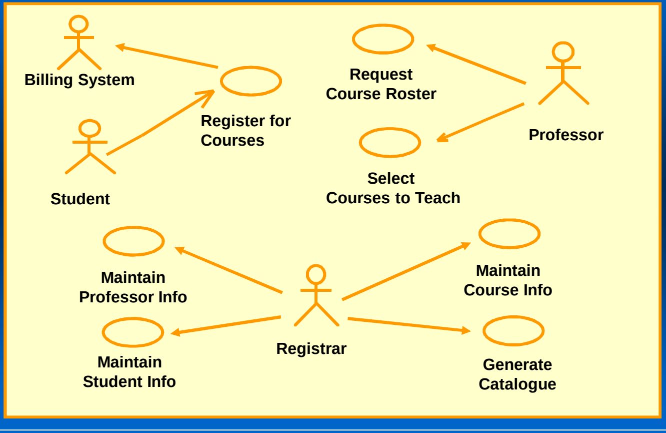
Build Visual Models



UPEDU Concept: Modeling Large Organization

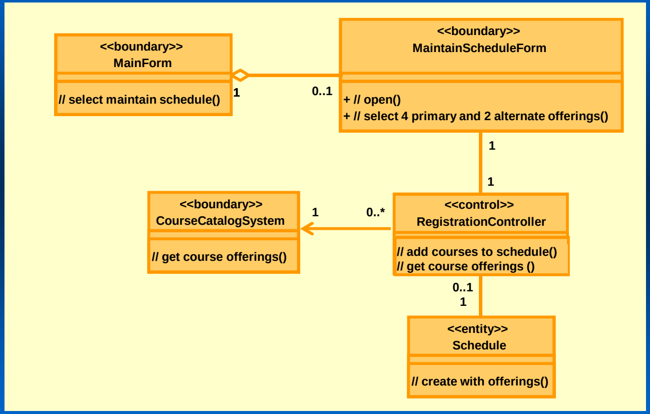
© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-13

Use Case Diagram



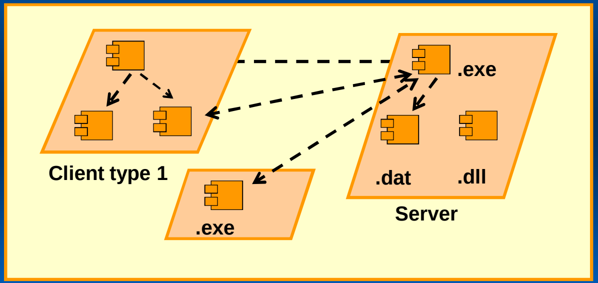
© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-14

Class Diagram

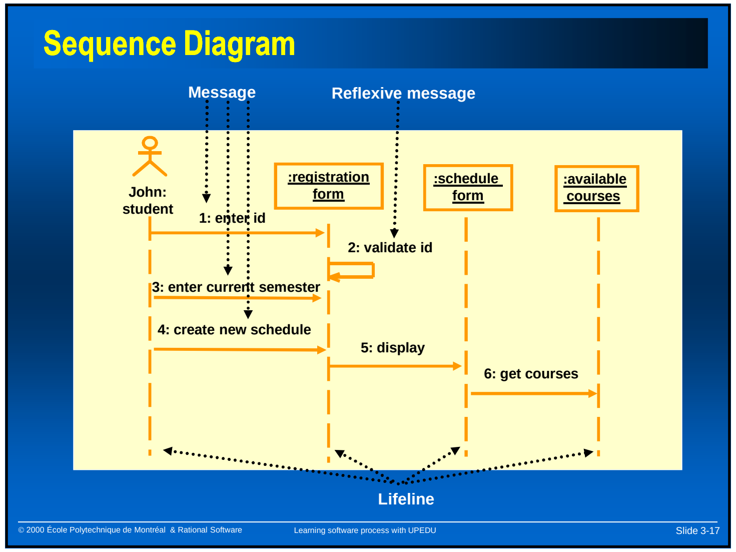

<<boundary>> MainForm
// select maintain schedule()
<<boundary>> MaintainScheduleForm
+ // open()
+ // select 4 primary and 2 alternate offerings()
1
0..1
1
1
<<boundary>> CourseCatalogSystem
// get course offerings()
1
0..*
<<control>> RegistrationController
// add courses to schedule()
// get course offerings ()
0..1
1
<<entity>> Schedule
// create with offerings()

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-15

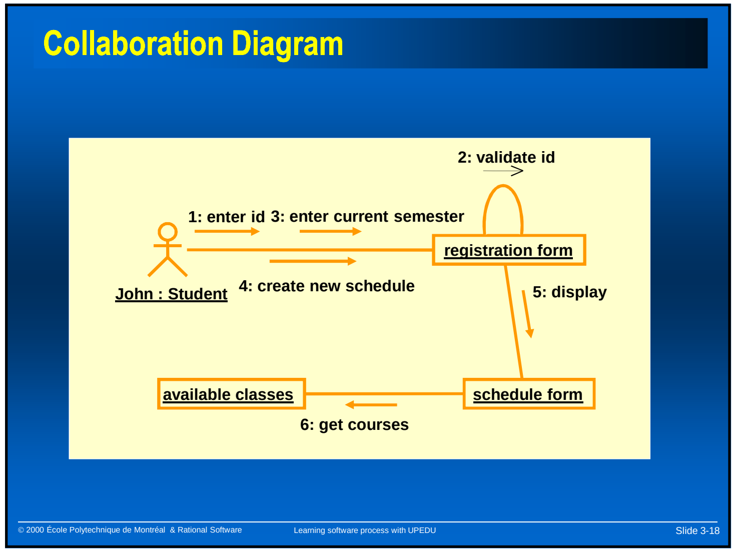
Component Diagram



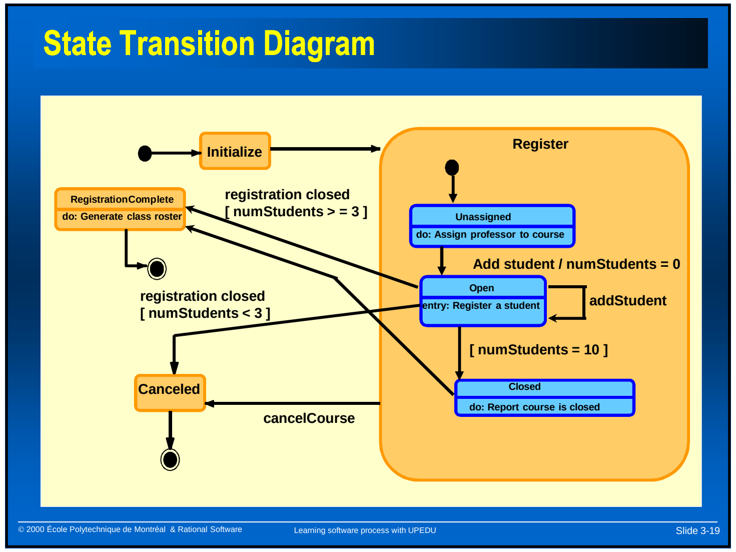
© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 3-16



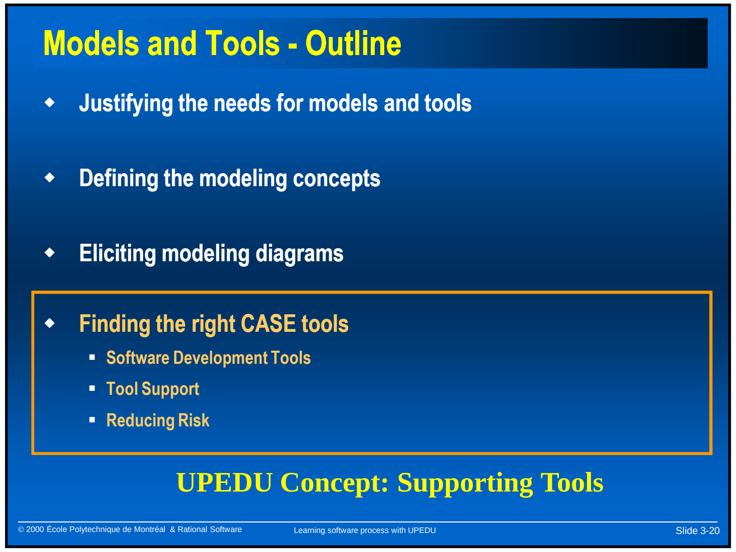
Sequence Diagram



Collaboration Diagram



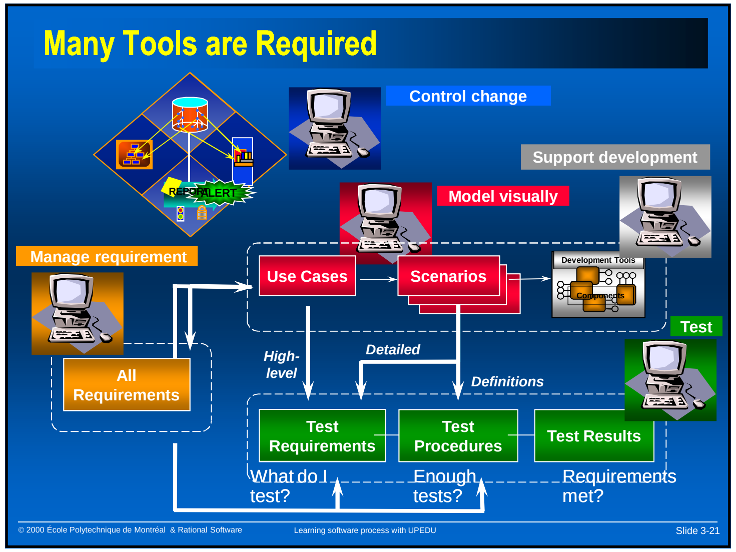
State Transition Diagram



Models and Tools - Outline

- Justifying the needs for models and tools
- Defining the modeling concepts
- Eliciting modeling diagrams
- Finding the right CASE tools
  - Software Development Tools
  - Tool Support
  - Reducing Risk

UPEDU Concept: Supporting Tools



Many Tools are Required

© 2000 École Polytechnique de Montréal & Rational Software   Learning software process with UPEDU   Slide 3-21