

# UPEDU : Unified Process for Education

## PARTIE 1 :

### Un exemple pour commencer

Principes : Processus de génie logiciel et cycle de vie du logiciel  
Modèles et outils

## Besoin d'un processus logiciel et Etude de cas

- ◆ **Spécification d'un projet d'une semaine**
  - Le cas d'étude
  - La vision
  - Le plan
  - Les facteurs de risque
- ◆ **Développement d'un projet d'une semaine**
- ◆ **Ce que l'on peut apprendre d'un processus de base d'ingénierie du logiciel**
- ◆ **Spécification d'une Etude de Cas**

## L'énoncé du problème

### *Le problème*

*Le problème de captage de données significatives sur le temps dépensé dans les différentes activités de développement logiciel qui affecte l'entreprise M dans sa capacité de gérer la progression d'un projet conformément à l'estimation, de facturer correctement le client, de payer les contractants et finalement d'obtenir une meilleure estimation du travail dans les projets futurs.*

- ◆ Les principaux acteurs sont :
  - Les développeurs individuels
  - L'assistant administratif
  - Les gestionnaires du projet

## L'énoncé de vision

### *L'énoncé de vision*

*For Martin's organization, a Personal Timing Tool (PTT) that measures time spent and collects and stores this data for later sorting and extraction- unlike "Post-it notes" and wild guesses- would enable Martin to make a systematic, consistent, and effortless assessment of where effort is spent; he could track actual versus estimated resources for a project; and he could do a better job planning for and developing future work.*

- ◆ Les cas d'utilisation :
  - Mesurer le temps pour chaque activité
  - Extraire les feuilles des temps hebdomadaires
  - Consolider des données pour un projet
  - Dégager un outil et une base de données for la détermination des ressources consacrées à un projet

### Plan de phases

Lundi	Mardi	Mercredi	Jeudi	Vendredi
Vision Plan Le cas d'étude Liste de risques	Prototype  Risques atténués	Conception Codage Test	Conception Codage Test	
Inception <i>Engagement de Martin</i>	Elaboration <i>Martin valide prototype</i>	Construction	Construction <i>Programmeur valide le produit logiciel</i>	
Prototype	Tests des cas d'utilisation	Conception Codage Test	Améliorations Transition : Livraison	

© 2000 École Polytechnique de Montréal & Rational Software      Learning software process with UPEDU      Traduction et adaptation par BTB      5

### Liste des risques

**Les risques**

**Environnement de développement**

**Environnement de l'utilisateur**

**Contraintes de coût**

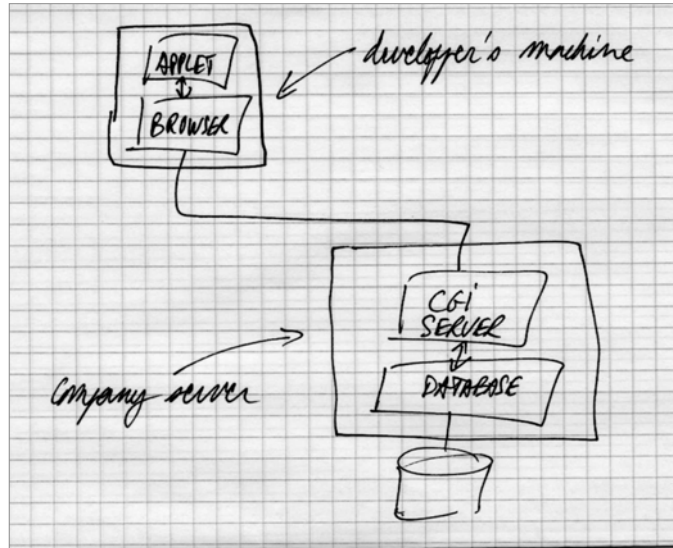
**Technologies du produit**

\_\_\_\_\_

\_\_\_\_\_

© 2000 École Polytechnique de Montréal & Rational Software      Learning software process with UPEDU      Traduction et adaptation par BTB      6

## Architecture



© 2000 École Polytechnique de Montréal &amp; Rational Software

Learning software process with UPEDU

Traduction et adaptation par BTD

7

## Besoins du processus logiciel et Etude de Cas

- ◆ Spécification d'un projet d'une semaine
- ◆ Développement d'un projet d'une semaine
  - Révision des besoins
  - Construction du prototype
  - Première exécution du produit logiciel
  - Beta Version
- ◆ Ce que l'on peut apprendre d'un processus de base d'ingénierie du logiciel
- ◆ Spécification d'une Etude de Cas

© 2000 École Polytechnique de Montréal &amp; Rational Software

Learning software process with UPEDU

Traduction et adaptation par BTD

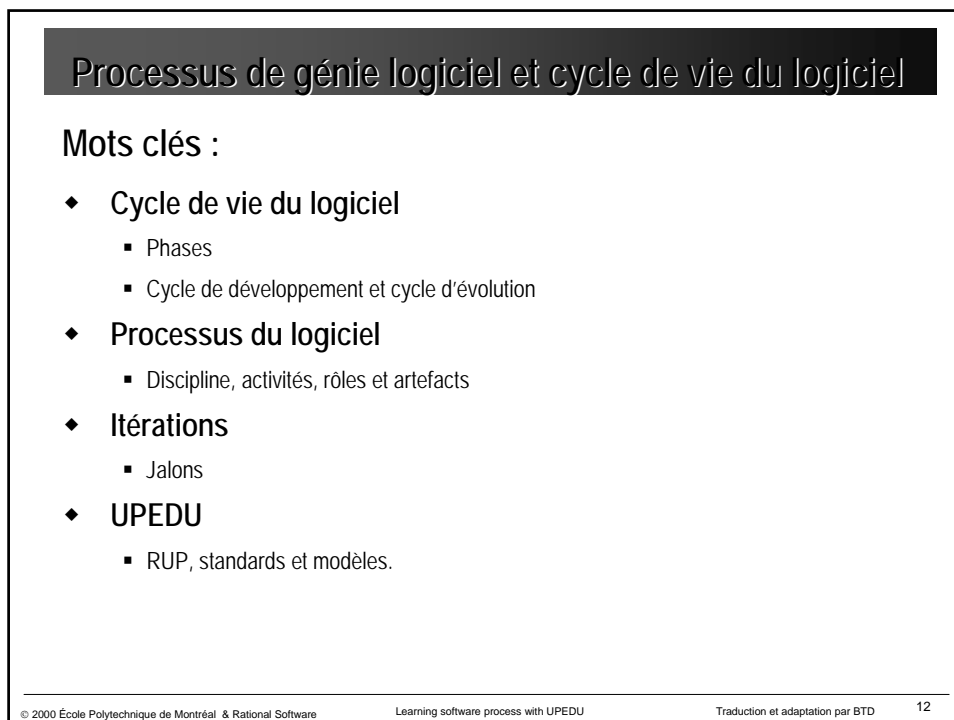
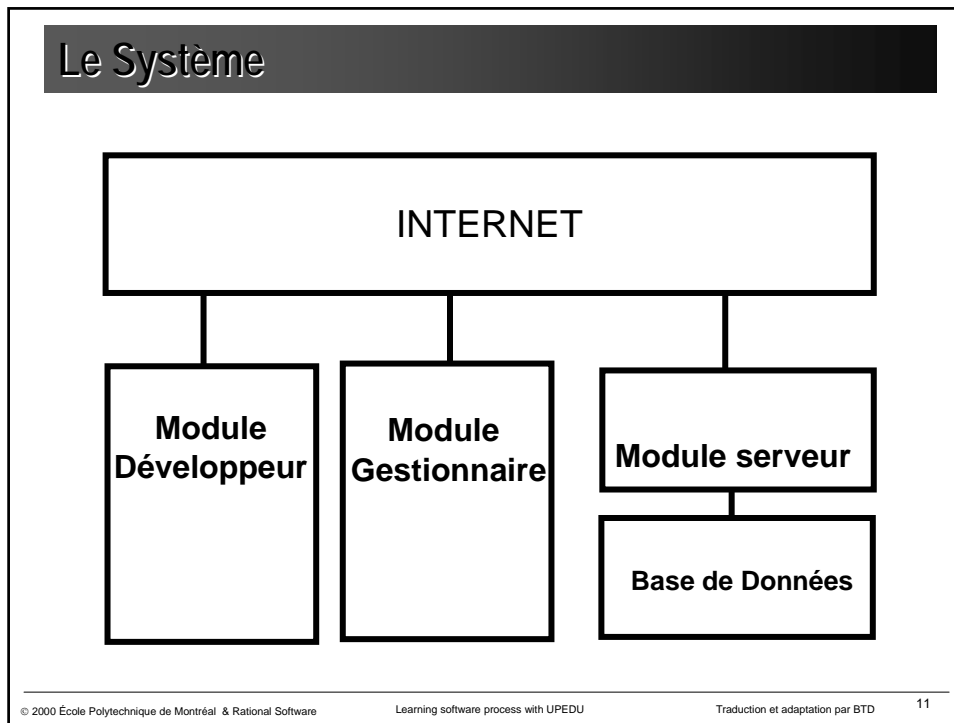
8

## Besoins du processus logiciel et Etude de Cas

- ◆ Spécification d'un projet d'une semaine
- ◆ Développement d'un projet d'une semaine
- ◆ Ce que l'on peut apprendre d'un processus de base d'ingénierie du logiciel
  - Mise en évidence du processus de base
  - Caractéristiques de l'équipe en charge du processus logiciel
  - Caractéristiques d'une organisation avec l'équipe importante
- ◆ Spécification d'une Etude de Cas

## Besoins du processus logiciel et Etude de Cas

- ◆ Spécification d'un projet d'une semaine
- ◆ Développement d'un projet d'une semaine
- ◆ Ce que l'on peut apprendre d'un processus de base d'ingénierie du logiciel
- ◆ Spécification d'une Etude de Cas
  - Introduction
  - Description globale
  - Besoins fonctionnels spécifiques



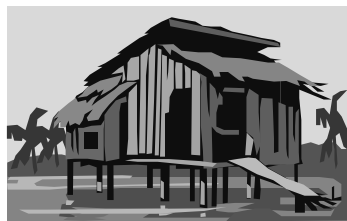
## Processus de génie logiciel et cycle de vie du logiciel

- ◆ Comprendre le Développement et l'évolution des logiciels
- ◆ Définition du cycle de vie du logiciel et du processus
- ◆ Définition des éléments du processus
- ◆ Intégration du Cycle de vie du logiciel et du processus du génie logiciel
- ◆ Adaptations du processus
- ◆ Récapitulatif

## Processus de construction de maisons



Même cycle de vie

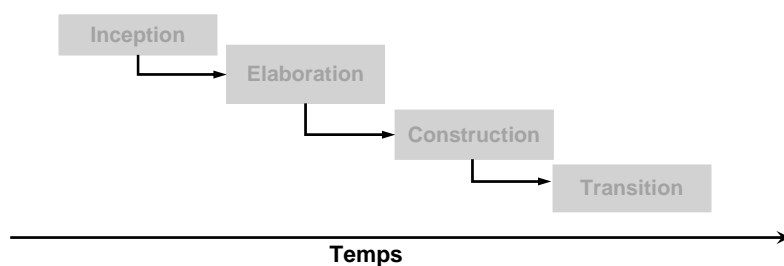


Différents Processus

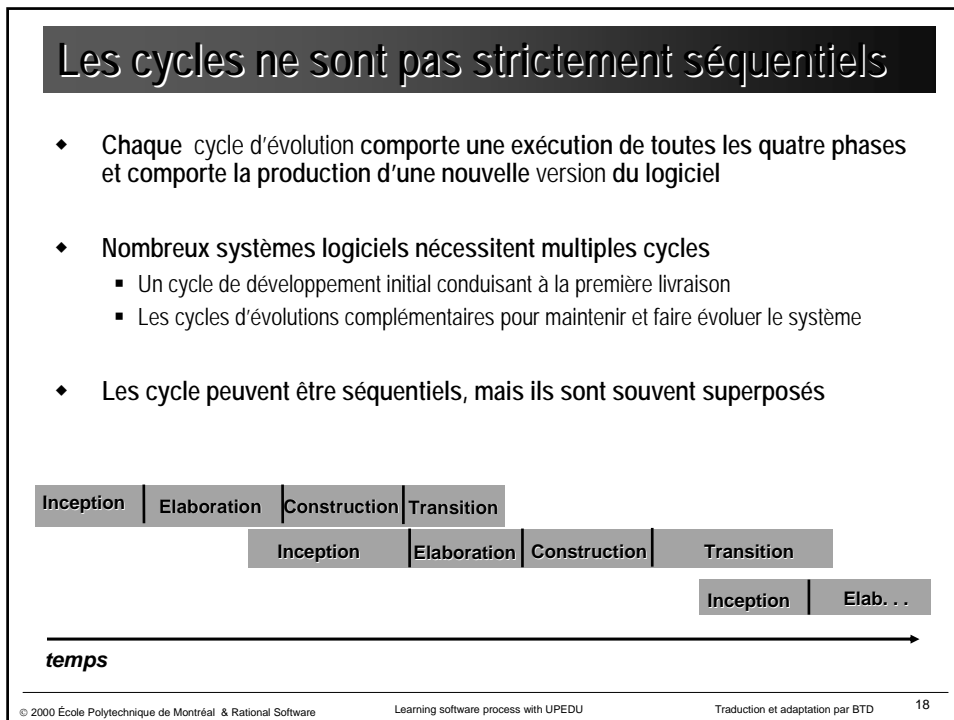
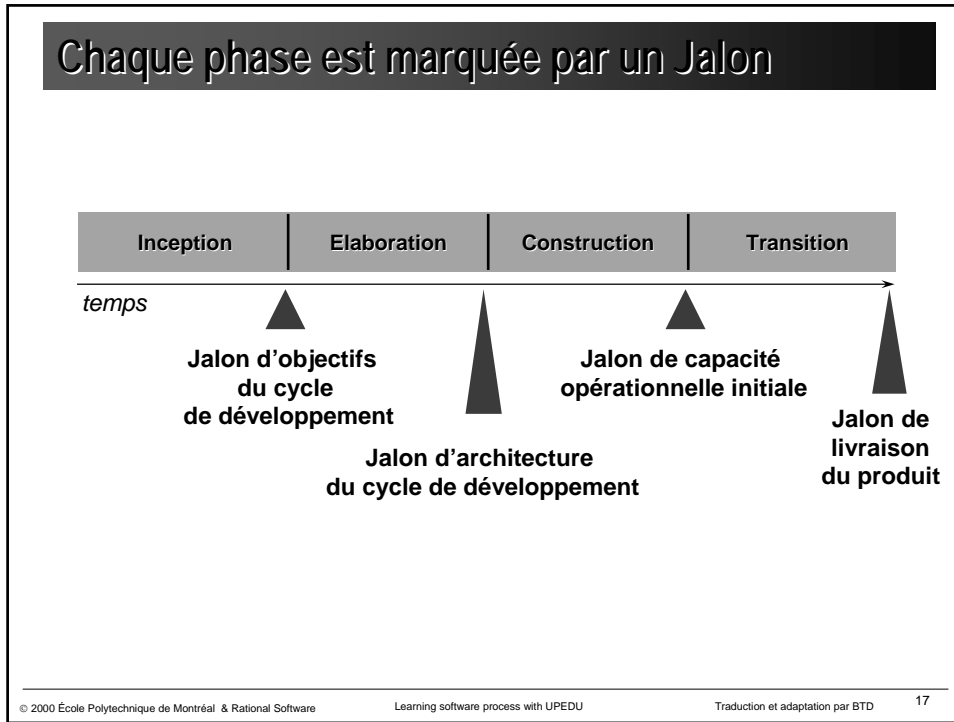
## Processus de génie logiciel et cycle de vie du logiciel

- ◆ Comprendre le Développement et l'évolution des logiciels
- ◆ Définition du cycle de vie du logiciel et du processus de génie logiciel
  - Cycle de vie du logiciel
  - Processus de génie logiciel
- ◆ Définition des éléments du processus de génie logiciel
- ◆ Intégration du Cycle de vie du logiciel et du processus de génie logiciel
- ◆ Adaptations du processus
- ◆ Récapitulatif

## Phases du Cycle de Développement



- Inception :** Définit l'étendu du projet
- Elaboration :** Planifie le projet, spécifie ses caractéristiques, fixe des fondements du système
- Construction :** Construit le produit
- Transition :** Transmet le produit dans la communauté des utilisateurs



## Qui travaille Quoi, Quand et Comment

```

        graph LR
            A[Des besoins nouveaux ou changés] --> B[Processus de Génie Logiciel]
            B --> C[Système nouveau ou modifié]
            style B fill:#ccc,stroke:#333,stroke-width:1px
            style A fill:none,stroke:none
            style C fill:none,stroke:none
            
```

**Fournir des indications pour un développement de logiciel efficace**

**Réduire les risques et augmente la prédictibilité**

**Capturer et présenter des “best practices”**

**Promouvoir la vision commune et une culture partagée**

© 2000 École Polytechnique de Montréal & Rational Software
Learning software process with UPEDU
Traduction et adaptation par BTD
19

## Disciplines d'Ingénierie et de Gestion

**Ingénierie**

-----

**Besoins**  
Requirements

**Analyse et Conception**  
Analysis & Design

-----

**Implementation**

-----

**Test**

Use-Case Model

Design Model

Implementation Model

Test Model

-----

**Gestion**  
Management

**Gestion de Configuration et de Changement**

-----

**Gestion de Projet**

Project Repository

© 2000 École Polytechnique de Montréal & Rational Software
Learning software process with UPEDU
Traduction et adaptation par BTD
20

## Processus de génie logiciel et cycle de vie du logiciel

- ◆ Comprendre le Développement et l'évolution des logiciels
- ◆ Définition du cycle de vie du logiciel et du processus

- ◆ Définition des éléments du processus
  - Modèle conceptuel de UPEDU
  - Rôles, Activités, Artefacts
  - Disciplines

- ◆ Intégration du Cycle de vie du logiciel et du processus
- ◆ Adaptations du processus
- ◆ Récapitulatif

© 2000 École Polytechnique de Montréal & Rational Software
Learning software process with UPEDU
Traduction et adaptation par BTD
21

## Notation

**Un rôle joué par un individu ou une équipe dans l'organisation de développement**

**Rôle**

**Spécifieur de Cas d'utilisation**

**Use Case**

**Use-Case Package**

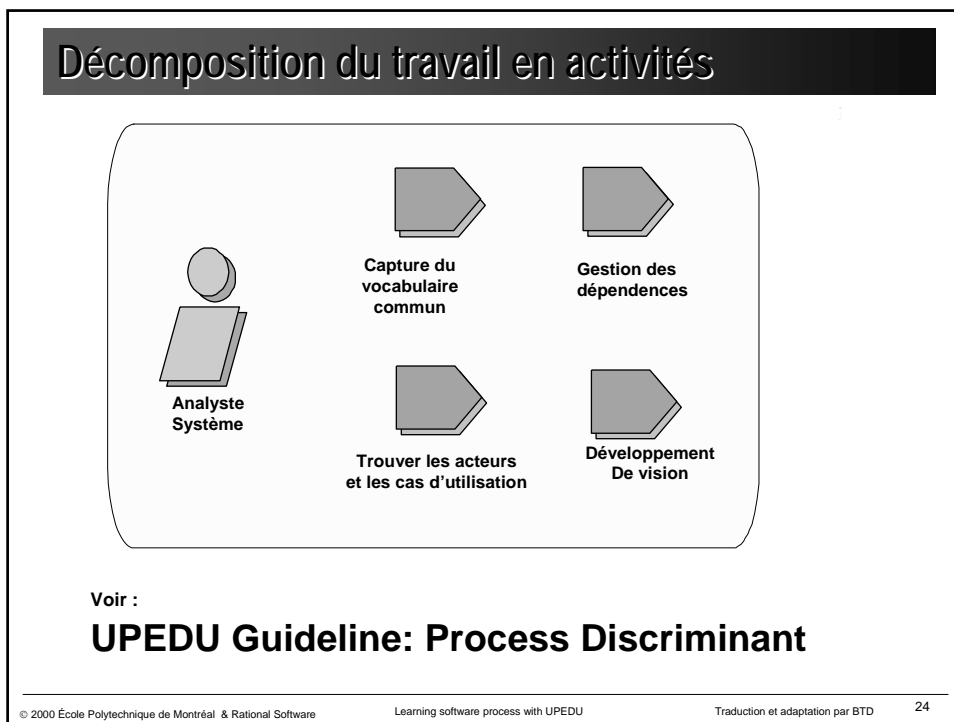
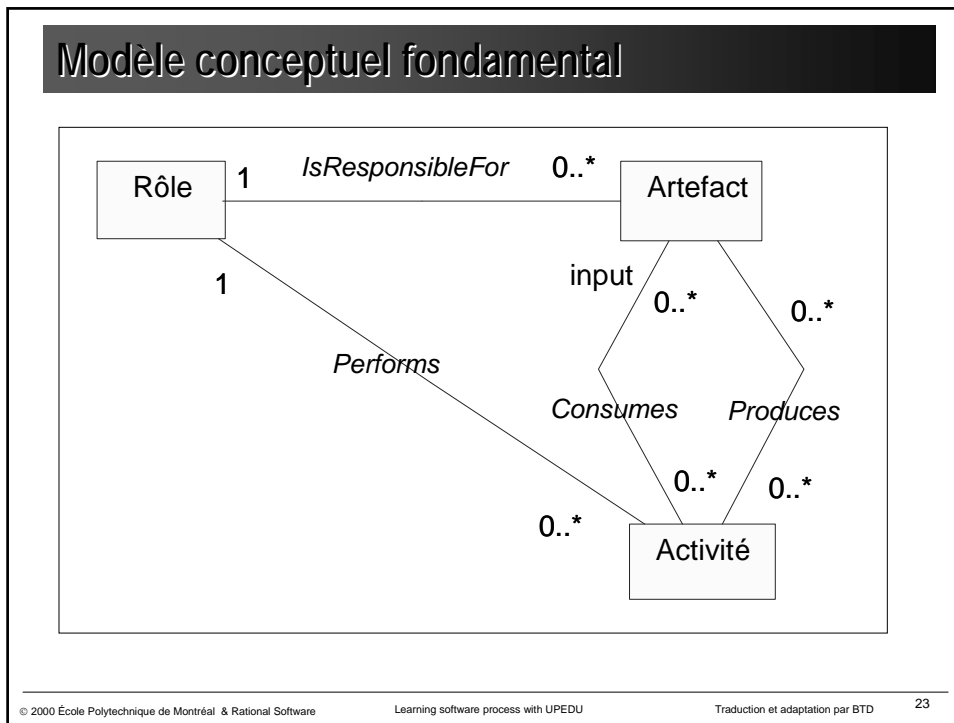
**Activité**

**Décrit un Use Case**

**Résultat**

**Une information produite, modifiée, ou utilisée par un processus**

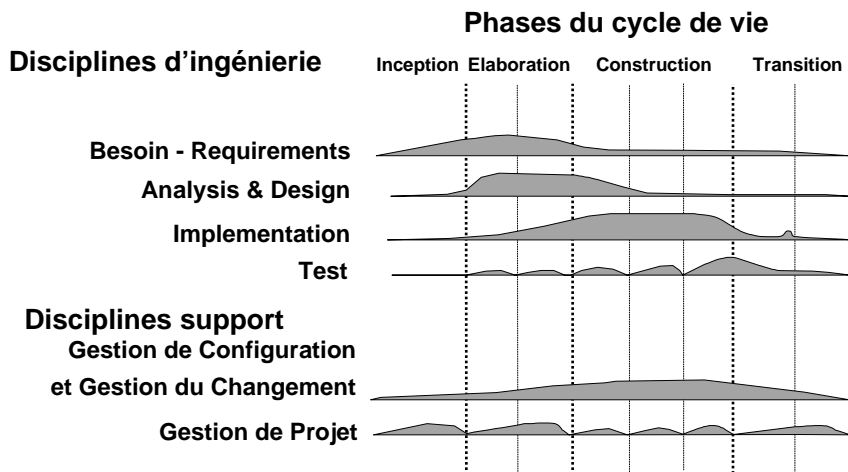
© 2000 École Polytechnique de Montréal & Rational Software
Learning software process with UPEDU
Traduction et adaptation par BTD
22



## Processus de génie logiciel et cycle de vie du logiciel

- ◆ Comprendre le Développement et l'évolution des logiciels
- ◆ Définition du cycle de vie du logiciel et du processus
- ◆ Définition des éléments du processus
- ◆ Intégration du Cycle de vie du logiciel et du processus
  - Disciplines et phases
  - Itérations
  - Activités systématiques et opportunistiques
- ◆ Adaptations du processus de génie logiciel
- ◆ Récapitulatif

## Modèle 2D du Processus et du cycle de vie





## Des itérations systématiques et opportunistes

**Possibles séquences opportunistes**

M : Management      G Gestion  
 R : Requirements    B Besoin  
 D : Design            C Conception  
 I : Implementation   I Implémentation

Systematic sequences  
**M R D I**

© 2000 École Polytechnique de Montréal & Rational Software      Learning software process with UPEDU      Traduction et adaptation par BTD      29

## Processus de génie logiciel et cycle de vie du logiciel

- ◆ Comprendre le Développement et l'évolution des logiciels
- ◆ Définition du cycle de vie du logiciel et du processus
- ◆ Définition des éléments du processus
- ◆ Intégration du Cycle de vie du logiciel et du processus de génie logiciel


- ◆ Adaptations du processus
  - Facteurs qui influent sur l'adaptation du processus
  - Standards et Modèles des processus de Génie Logiciel

- ◆ Récapitulatif


© 2000 École Polytechnique de Montréal & Rational Software      Learning software process with UPEDU      Traduction et adaptation par BTD      30

## Quatre facteurs d'influence


**Cycle de vie du produit**




**Domaine d'application**



**Processus**



**Personnes et Organisation**



**Voir :  
UPEDU Concept: Tailoring the Process**

© 2000 École Polytechnique de Montréal & Rational Software
Learning software process with UPEDU
Traduction et adaptation par BTB 31

## Standards et Modèles du processus de Génie Logiciel

- ◆ **IEEE std 1074-1995 processes**
  - Gestion de projet
  - Pre-développement
  - Développement
  - Post-développement
  - Intégral
- ◆ **Modèles**
  - Cascade (Waterfall)
  - Spiral
  - Incrémental
  - Prototypage

**Voir :  
UPEDU Concept: Effects of implementing a Process**

© 2000 École Polytechnique de Montréal & Rational Software
Learning software process with UPEDU
Traduction et adaptation par BTB 32

## Processus de génie logiciel et cycle de vie du logiciel

- ◆ Comprendre le Développement et l'évolution des logiciels
- ◆ Définition du cycle de vie du logiciel et du processus
- ◆ Définition des éléments du processus
- ◆ Intégration du Cycle de vie du logiciel et du processus de génie logiciel
- ◆ Adaptations du processus

- ◆ Récapitulatif

## Synthèse

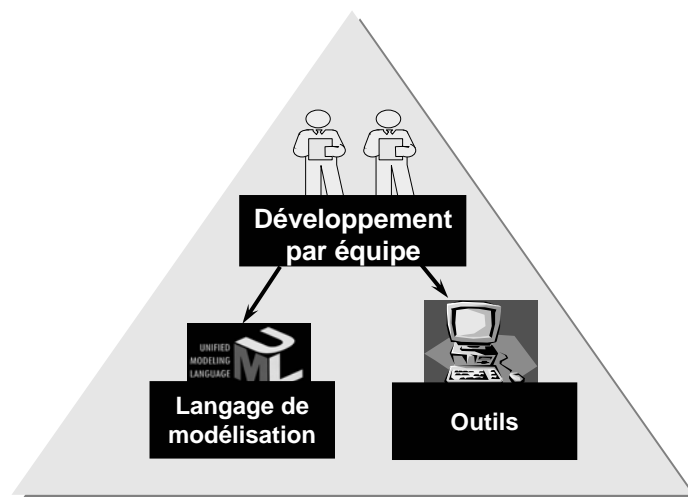
- ◆ Processus de Génie Logiciel versus Cycle de vie du logiciel
- ◆ Les phases d'un cycle de vie de logiciel
- ◆ Les disciplines du processus de Génie Logiciel
- ◆ Les composants des disciplines
- ◆ Cycle de développement vs cycle d'évolution
- ◆ Interaction entre les disciplines et les phases
- ◆ Itérations et processus de Génie Logiciel
- ◆ Différents modèles et standards des processus de Génie Logiciel et des cycles de vie de logiciel

## Modèles et Outils

- ◆ Justifier les besoins de modèles et d'outils
  - La plupart des projets d'ingénierie comptent sur les techniques de modélisation et des outils associés pour faciliter le développement des projets. UML est le langage de modélisation conçu spécifiquement pour faciliter la modélisation de systèmes logiciels sans référence à l'implémentation envisagée. L'efficacité du processus de génie logiciel est associée à l'utilisation des outils de GLAO (Génie Logiciel Assisté par Ordinateur).
- ◆ Définir des concepts de modélisation
- ◆ Expliciter des diagrammes de modélisation
- ◆ Trouver des outils GLAO appropriés

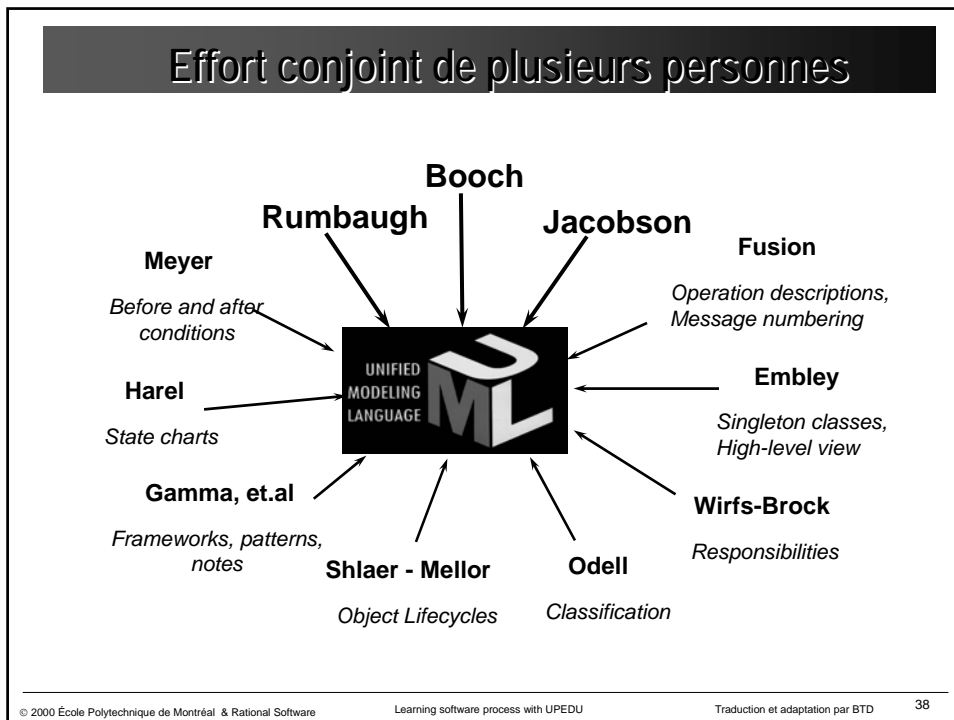
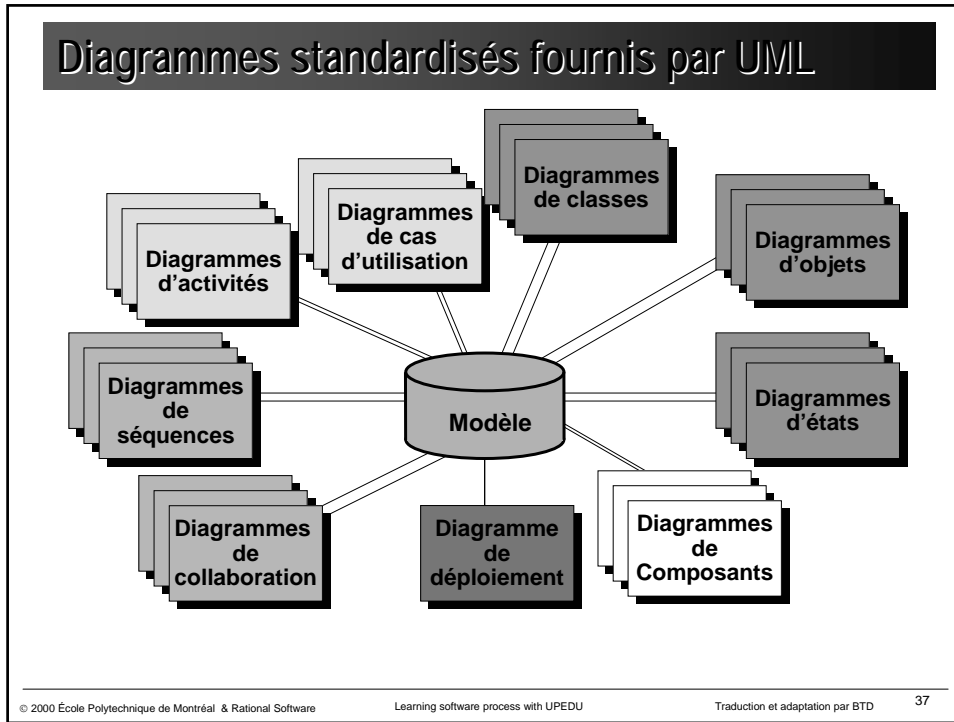
CASE – Computer Aided Software Engineering  
GLAO – Génie Logiciel Assisté par Ordinateur

## Les ingrédients de succès pour le processus de Génie Logiciel



Voir :

**UPEDU: Best Practice: Model Visually**



## Modèles et Outils

- ◆ Justifier les besoins de modèles et d'outils

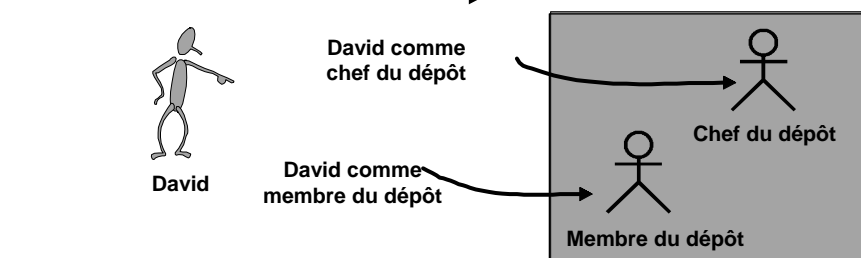
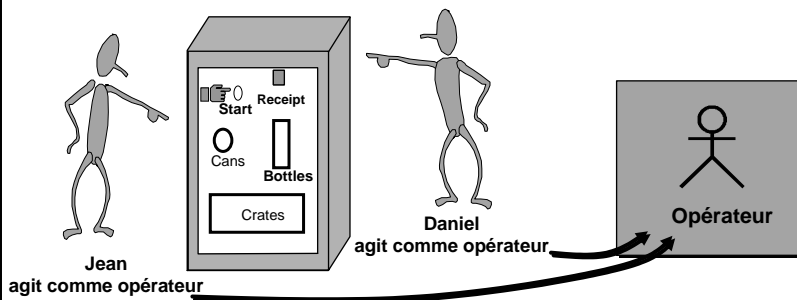
- ◆ Définir des concepts de modélisation

- Acteur
- Cas d'utilisation (Use-Case)
- Classes
- Associations
- Composants and Packages

- ◆ Expliciter des diagrammes de modélisation

- ◆ Trouver des outils GLAO appropriés

## La différence entre un acteur et un individu



## Le modélisateur établit les frontières

**Frontières du système ?**

Le répondeur est-il un acteur ou une partie du système ?

**Un acteur ne fait pas partie du système**

**Un acteur peut être :**

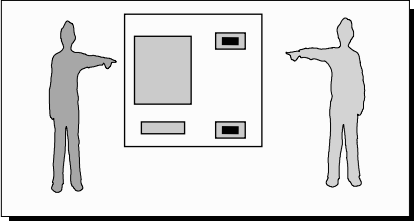
- un **humain**,
- une **machine** ou
- un **autre système**

**Un acteur échange les informations avec le système :**

- **donne les informations**
- **reçoit les informations**

© 2000 École Polytechnique de Montréal & Rational Software      Learning software process with UPEDU      Traduction et adaptation par BTD      41

## Chaque cas d'utilisation est initié par un acteur



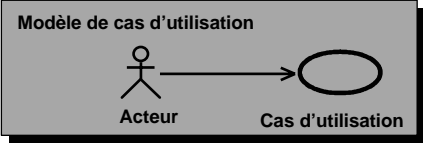
**Un cas d'utilisation est une séquence d'actions que le système effectue en produisant des résultats observables et significatifs pour un acteur particulier.**

**Un cas d'utilisation modélise les dialogues entre les acteurs et le système.**

**Un cas d'utilisation exprime le flux complet et significatif des événements.**

**Mis ensemble tous les cas d'utilisation constituent toutes les façons possibles d'utiliser le système.**

**Modèle de cas d'utilisation**




Acteur                      Cas d'utilisation


© 2000 École Polytechnique de Montréal & Rational Software      Learning software process with UPEDU      Traduction et adaptation par BTD      42


## La notation pour déclarer des Classes

- ♦ Une classe est décrite par trois sections
  - Nom de la classe, sa structure (attributs), son comportement (operations)
- ♦ Une entité classe modélise des associations et des informations permanentes (persistantes)
  - Phénomène réel, les tâches internes au système, les valeurs d'attributs qui sont souvent fournies par un acteur
- ♦ La classe de frontière modélise la communication entre le système et son environnement
  - Windows (interface utilisateur), Protocole de communication (interface système)
- ♦ La classe de contrôle modélise le comportement de contrôle spécifique a une ou plusieurs cas d'utilisation
  - Créer, initialiser, effacer, séquencer, coordonner l'exécution et contrôler des objets

<b>Professeur</b>
nom
empID
créer()
sauver()
effacer()
changer()

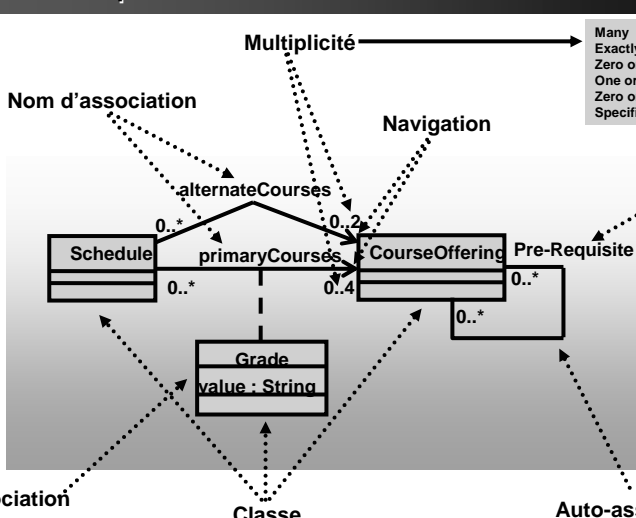






© 2000 École Polytechnique de Montréal & Rational Software      Learning software process with UPEDU      Traduction et adaptation par BTD      43

## Associations représentent des relations structurelles



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

Nom d'association

Multiplicité

Navigation

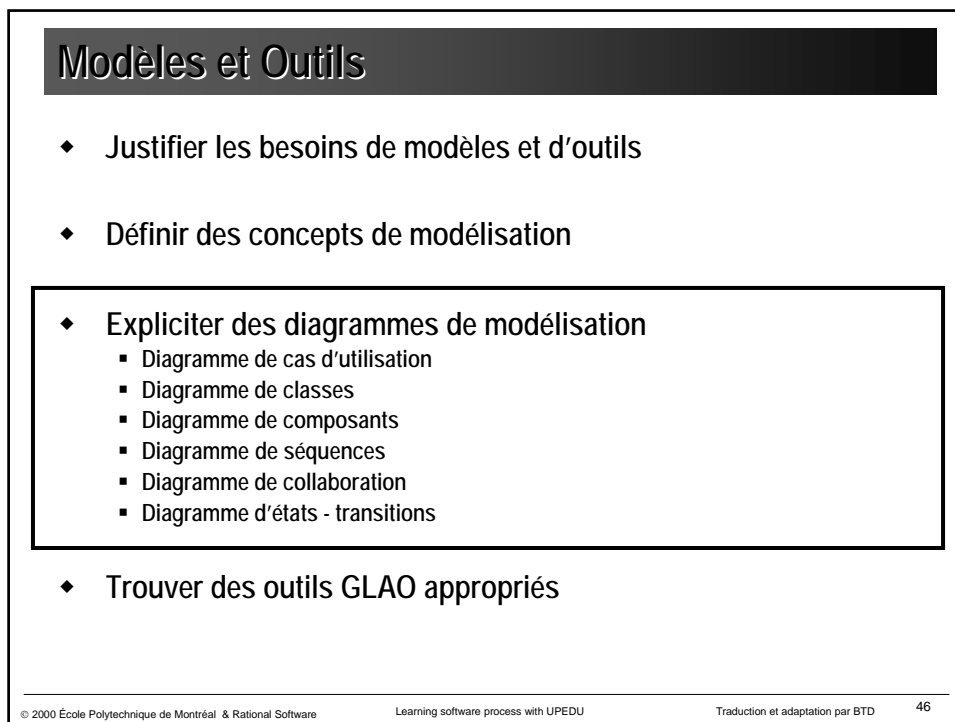
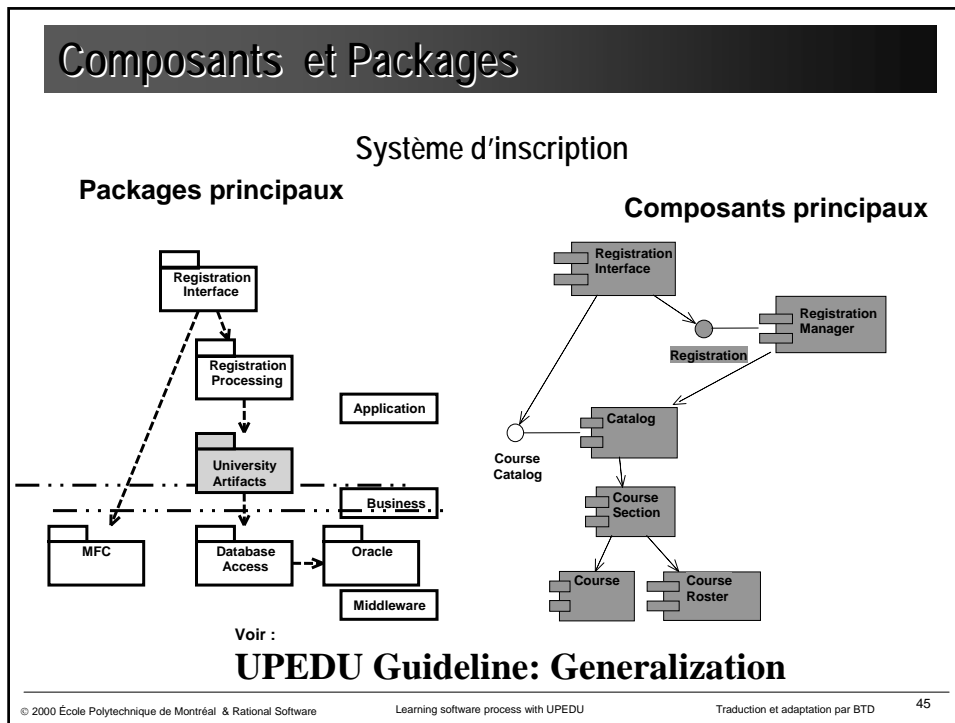
Nom du rôle

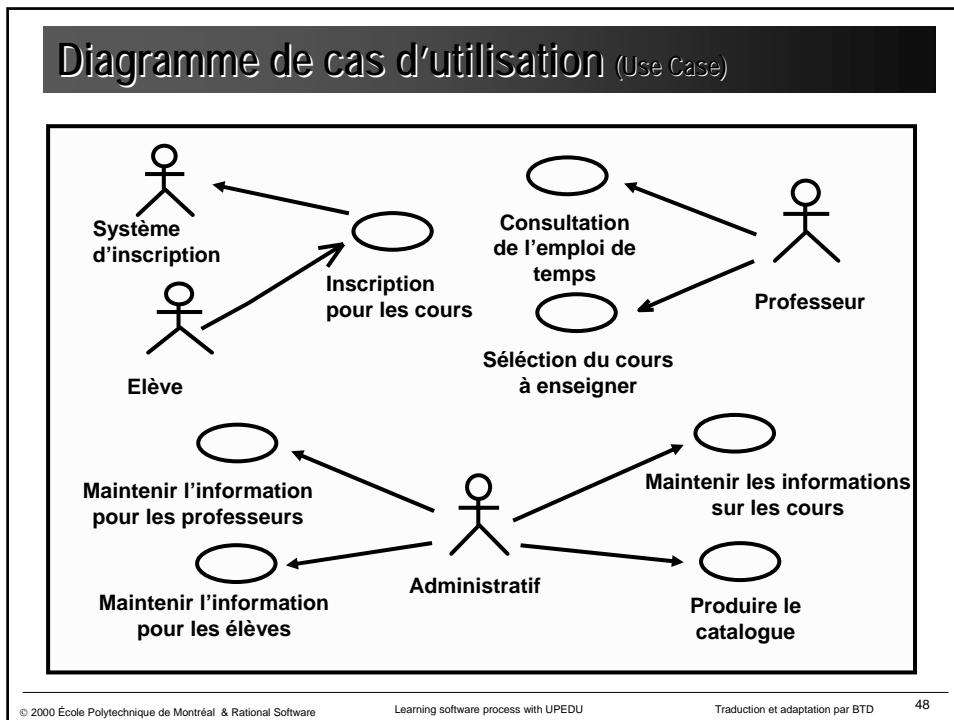
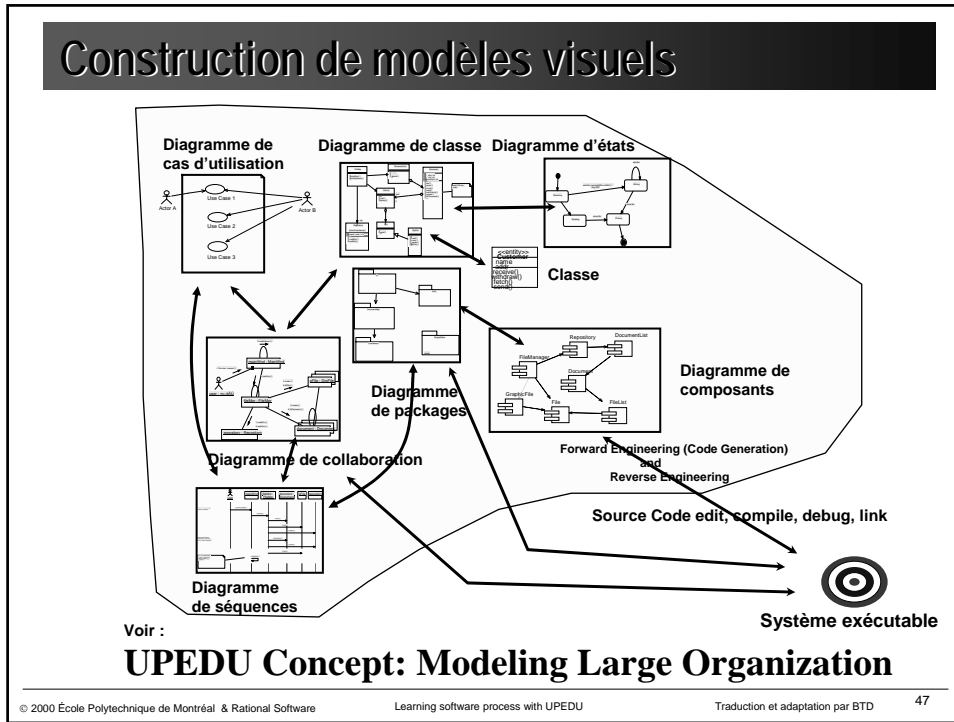
Classe d'association

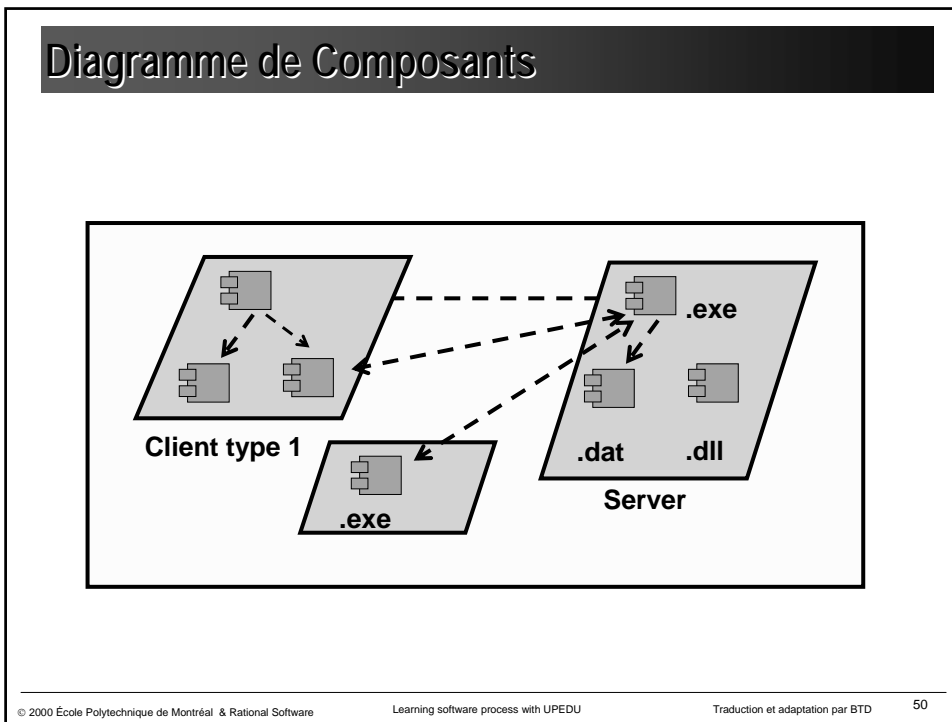
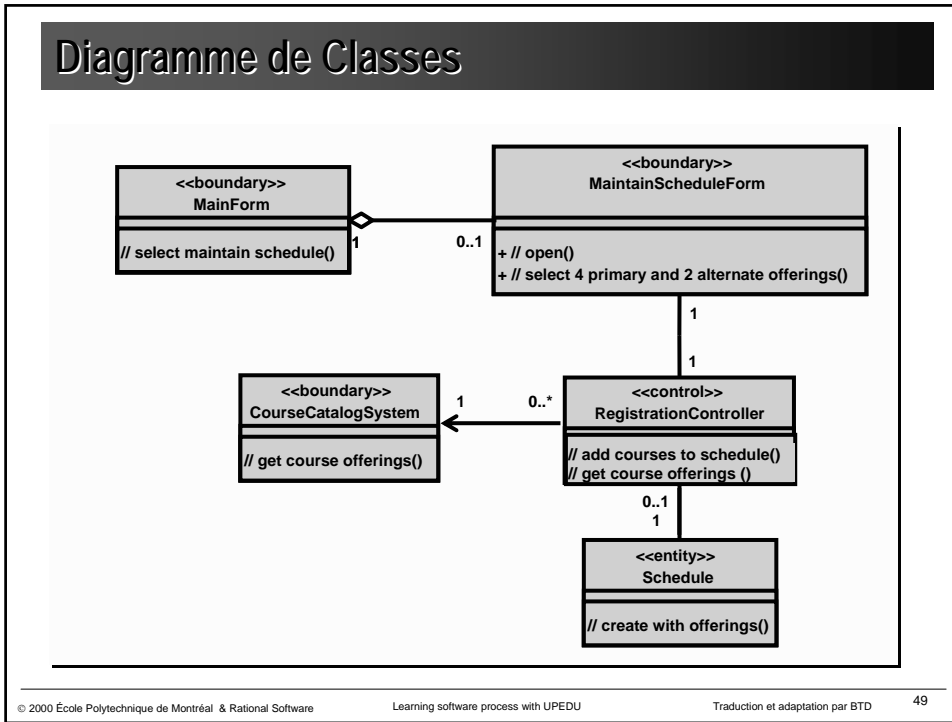
Classe

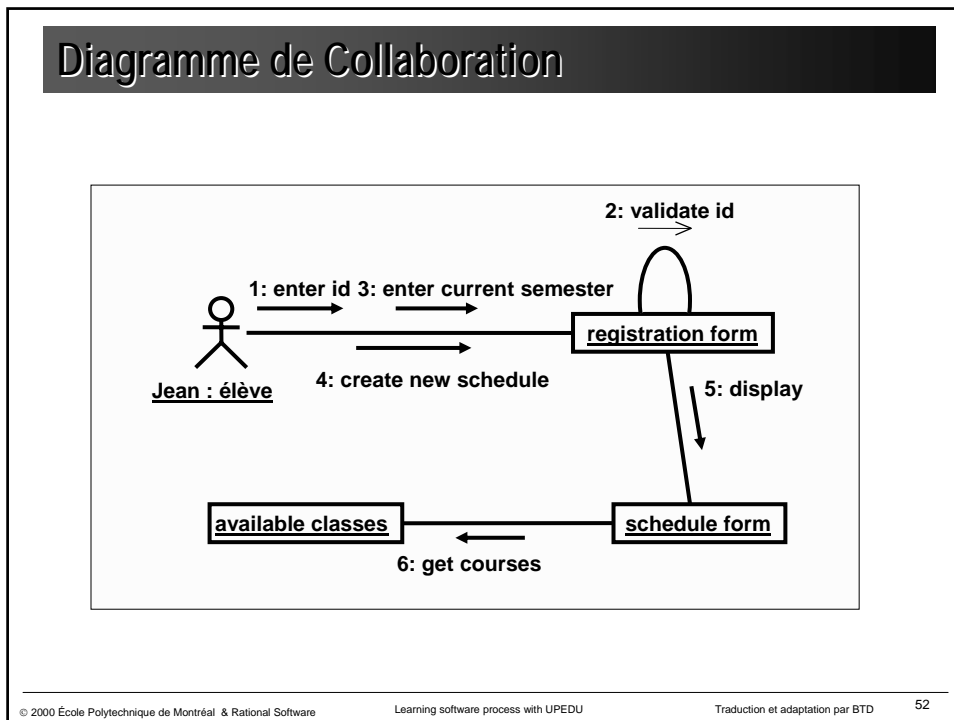
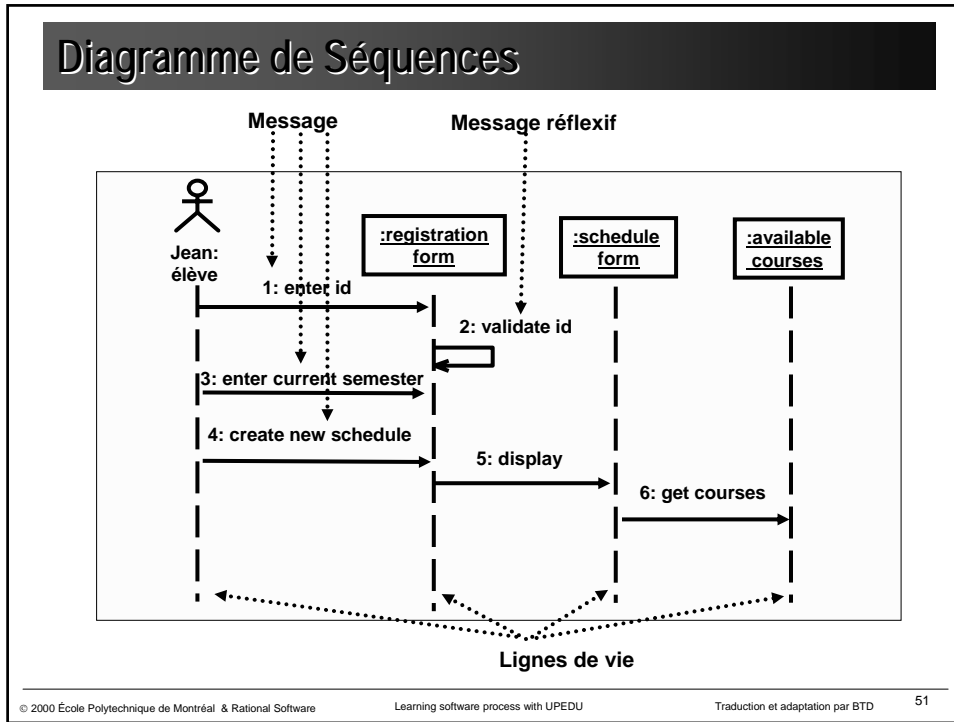
Auto-association

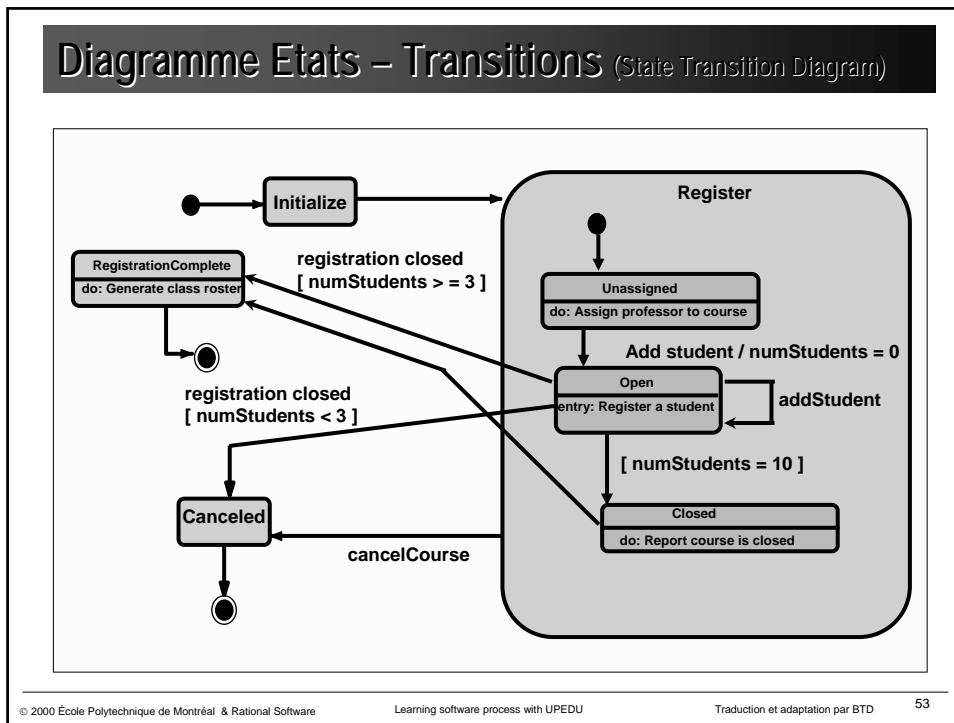
© 2000 École Polytechnique de Montréal & Rational Software      Learning software process with UPEDU      Traduction et adaptation par BTD      44











## Modèles et Outils

- ◆ Justifier les besoins de modèles et d'outils
- ◆ Définir des concepts de modélisation
- ◆ Expliciter des diagrammes de modélisation

- ◆ Trouver des outils GLAO appropriés
  - Software Development Tools
  - Tool Support
  - Reducing Risk

Voir :

**UPEDU Concept: Supporting Tools**

