

Introduction à UML

Introduction à la Spécification

Plan du cours

- 1. Introduction à la Spécification**
2. Aperçu d'UML
3. Diagramme de Use Case et Description Textuelle
4. Diagramme de Classes
5. Diagramme d'Activité
6. Diagramme d'États
7. Diagramme de Séquence

Qu'est ce qu'une Spécification ?

Une **spécification** est la description des fonctions d'un logiciel en vue de sa réalisation.

Permet de définir :

- le **QUOI** : ce que doit faire le système.

MAIS

- pas le **COMMENT** : pas de prescription sur la manière dont le système réalisera ses fonctions.

Niveaux de spécification

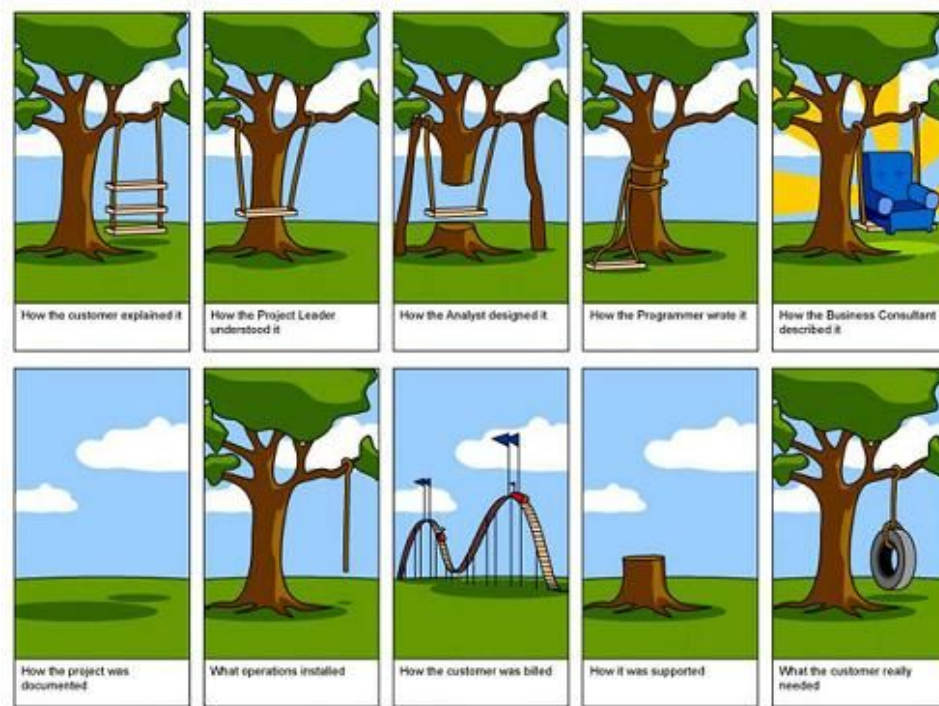
1. Analyse

- Analyser le domaine d'application du système.
- Comprendre les besoins.

2. Conception

- Concevoir le système en termes de classes et objets.
- Modéliser une solution.

Problème de communication



Types de Langages de Spécification

1. Langage naturel
2. Langage formel
3. Langage semi-formel

Langage naturel

Langue « *normale* » parlée par un être humain

- Avantages
 - Pas (peu) d'apprentissage.
- Inconvénients
 - Ambigu.
 - Manque de visibilité.
 - ...
- Exemples
 - Français utilisé dans les descriptions textuelles des Cas d'Utilisation.

Langage naturel

Use Case 2 🏠 Get Paid for Car Accident ✍️

Primary Actor: Claimant

Scope: Insurance company ("MyInsCo")

Level: Summary

Stakeholders and Interests:

Claimant—to get paid the most possible.

MyInsCo—to pay the smallest appropriate amount.

Department of Insurance—to see that all guidelines are followed.

Precondition: None.

Minimal Guarantees: MyInsCo logs the claim and all activities.

Success Guarantees: Claimant and MyInsCo agree on amount to be paid; claimant gets paid that.

Trigger: Claimant submits a claim.

Main Success Scenario:

1. Claimant submits claim with substantiating data.
2. Insurance company verifies claimant owns a valid policy.
3. Insurance company assigns agent to examine case.
4. Insurance company verifies all details are within policy guidelines.
5. Insurance company pays claimant and closes file.

Extensions:

1a. Submitted data is incomplete:

1a1. Insurance company requests missing information.

1a2. Claimant supplies missing information.

2a. Claimant does not own a valid policy:

2a1. Insurance company denies claim, notifies claimant, records all this, terminates proceedings.

3a. No agents are available at this time.

3a1. (What does the insurance company do here?)

4a. Accident violates basic policy guidelines:

4a1. Insurance company denies claim, notifies claimant, records all this, terminates proceedings.

4b. Accident violates some minor policy guidelines:

4b1. Insurance company begins negotiation with claimant as to amount of payment to be made.

Langage formel

Langage avec une syntaxe et une sémantique précise

- Avantages
 - Pas d'ambiguïté.
- Inconvénients
 - Difficile à apprendre.
 - Expressivité souvent réduite.
- Exemples
 - Logique des prédicat, logique temporelle, Z.
 - Code: PHP, C++, C#, Java...

Langage formel



Objectifs de la formation

Introduction à la spécification

- Comprendre l'importance d'une spécification.
- Savoir rédiger une spécification.

Introduction à la conception OO

- Comprendre les mécanismes OO.
- Pouvoir structurer un programme en termes de classes et d'objets.

Plan du cours

1. Introduction à la Spécification
- 2. Aperçu d'UML**
3. Diagramme de Use Case et Description Textuelle
4. Diagramme de Classes
5. Diagramme d'Activité
6. Diagramme d'États
7. Diagramme de Séquence
8. Autres Diagrammes

U - M - L

Unified

- Ensemble de diagramme (complétude)

Modeling

- Modèle, vue abstraite et cohérente d'une problématique

Language

- Outils de communication semi-formel
- Définit des règles de construction (correction)

Les différentes perspectives

1. Perspective informationnelle

- Contenu: les données.
- Diagramme de classes UML (autre: Entité-Association).

2. Perspective fonction

- Contenu: les services, tâches, opérations,...
- Diagramme de Use Case UML (autre: Features Diagrams).

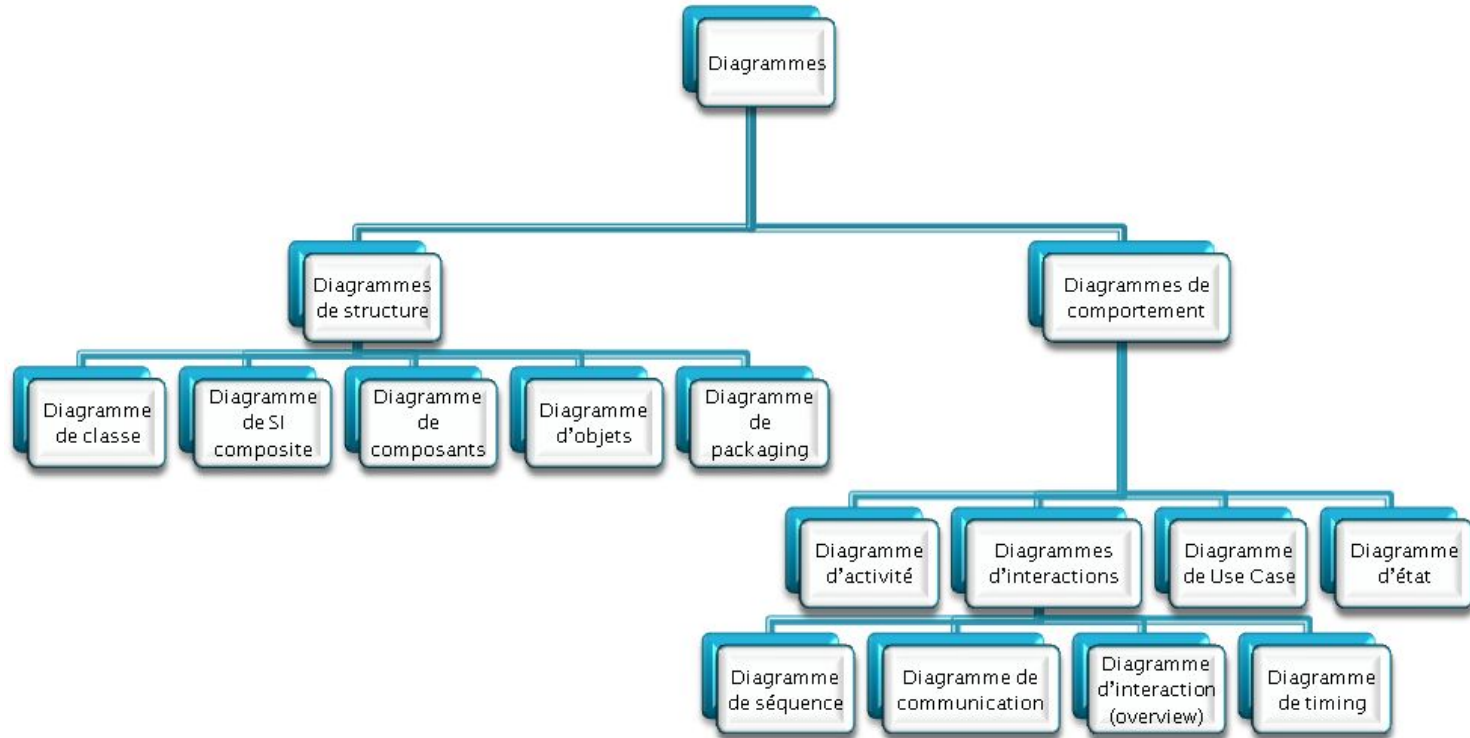
3. Perspective dynamique

- Contenu: le comportement des objets, acteurs, document,...
- Diagramme de séquence, de collaboration, d'état et d'activité UML (autre: réseaux de Petri).

4. Perspective de l'Orienté Objet

- Contenu: données + fonctions (dynamisme!)
- Diagramme de Classe UML

Les différents diagrammes



Correction et Cohérence

Quand un diagramme est-il bon?

1. **Correct**

Respecte les règles du langage (de modélisation)

2. **Cohérent**

Modéliser adéquatement le domaine d'application

Justesse et Précision

Un bon diagramme UML n'est pas forcément le plus détaillé. Il faut adapter le niveau de détails en fonction de l'objectif de la modélisation (analyse / conception).

Quelle valeur de Pi vaut-il mieux utiliser ?

4.14159265358979323846264

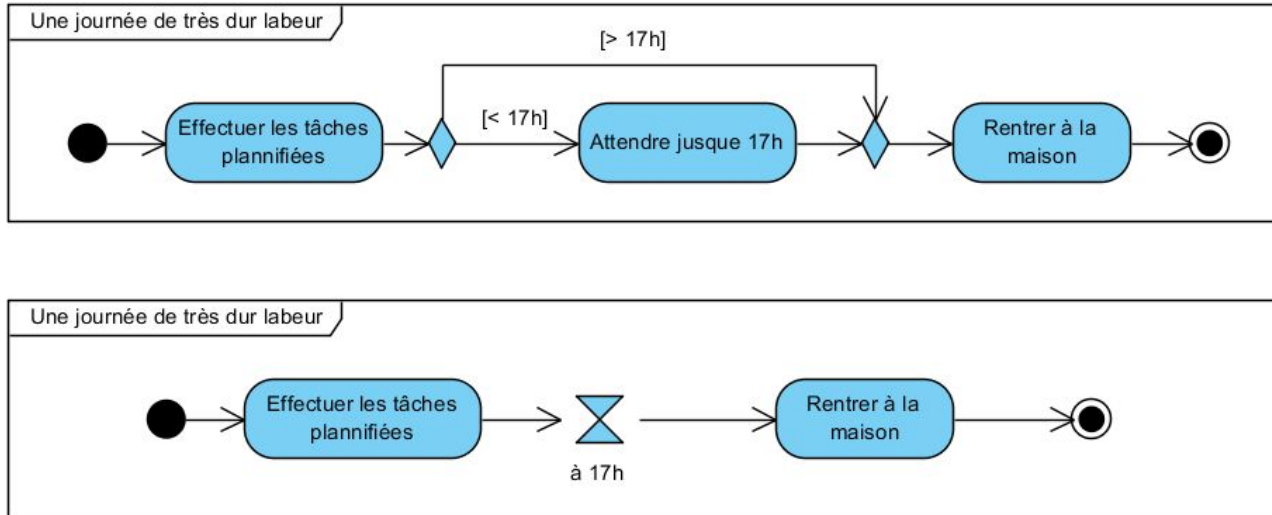
(précise mais pas juste)

ou

3.1

(moins précise mais plus juste)

Justesse et Précision



Complétude

Est-il possible de décrire complètement ce bâtiment au travers d'un plan unique ?



Complétude

Est-il possible de décrire complètement ce bâtiment au travers d'un plan unique ?



Vue de l'architecte



Vue de l'électricien



Vue de l'urbaniste



Vue du client