Conception de systèmes d'information

Cours 5 – UML

František Kardoš

fkardos@labri.fr

UML – Unified Modeling Language

UML est une langage graphique qui permet de représenter et de communiquer les divers aspects d'un système d'information (SI).
Aux graphiques sont bien sûr associés des textes qui expliquent leur contenu.
UML comporte de différents types de diagrammes représentant des vues distinctes pour représenter des concepts particuliers du SI.

Diagrammes structurels / statiques

- diagramme de classes
- diagramme d'objets
- diagramme de composants
- diagramme de déploiement
- diagramme de paquetages
- diagramme de structures composites

Diagrammes comportementaux / dynamiques

- diagramme de cas d'utilisation
- diagramme d'activités
- diagramme d'état-transitions
- diagrammes d'interaction
  - diagramme de séquence
  - diagramme de communication
  - diagramme global d'interaction
  - diagramme de temps

UML vs MERISE

MERISE est une méthodologie :

- démarche (étapes, phases, tâches)
- formalismes (techniques de modélisation)

UML est un langage "unificateur" de formalismes.

UML vs MERISE

MERISE est une méthodologie :

- démarche (étapes, phases, tâches)
- formalismes (techniques de modélisation)

UML est un langage "unificateur" de formalismes.

Ils sont donc complémentaires : on peut profiter des formalismes de UML et de méthodes de MERISE dans le même projet.

UML vs MERISE

Modélisation de traitements de MERISE est orienté processus

UML est plutôt adapté pour des modèles orientés objet

UML vs MERISE

Cycle de vie MERISE : analyse, conception, implémentation.

Cycle de vie itératif : analyse + conception + implémentation d'un prototype, développement de fonctionnalités/modules un(e) par un(e).

En théorie, le choix de logiciel se fait très tard ; dans la pratique ce choix est fait bien avant, en raison des contraintes de coûts, des systèmes déjà mis en place, des "habitudes", etc.

Diagramme de cas d'utilisation

- la première étape UML d'analyse d'un système
- moyen simple de recenser les grandes fonctionnalités d'un système.

Éléments des diagrammes :

- acteur
- cas d'utilisation

Acteur

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système.

Représenatation graphique : un petit bonhomme, portant le nom.

Cas d'utilisation

Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur.

Représentation graphique : une ellipse, portant le nom.

Relations entre cas d'utilisation

- Dépendances : inclusion, extension
  - une flèche simple, un trait pointillé
- Généralisation / spécialisation
  - une flèche triangle, un trait plein

Description d'un cas d'utilisation

- nom
- objectif (résumé)
- acteurs principaux, acteurs secondaires
- date de création, version, responsable

- les préconditions
- des scénarii (le scénario nominal, les scénarii d'exception – cas d'erreurs)
- des postconditions

Classes et objets

La notion de classe de UML correspond (grosso-modo) à la notion d'entité de MERISE. Un objet correspond alors à une occurence d'une entité.

Une *classe* est la description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communes.

Un *objet* est une instance d'une classe. C'est une entité concrète dotée d'une identité, d'un état et d'un comportement que l'on peut invoquer.

Diagramme de séquence

Un diagramme de séquence décrit les messages échangés entre les acteurs et les objets, présentés par les lignes de vie, dans un ordre chronologique.

Le temps est représenté explicitement par la dimension verticale et s'écoule de haut en bas.

Élements des diagrammes :

- ligne de vie
- message synchrone / asynchrone
- message de création / destruction
- ...

Ligne de vie

est une ligne verticale, avec comme étiquette, le nom pour un acteur, et pour un objet le nom de rôle et/ou le nom de type.

```
[<nom_du_rôle>] : [<nom_du_type>]
```

Message asynchrone

Généralement, c'est l'envoie d'un signal, un événement ou une interruption.
L'émetteur du message n'attend pas de réponse et il n'est pas bloqué pendant l'envoie du message.

Graphiquement, un message asynchrone se représente par une flèche en traits pleins et à l'extrémité ouverte partant de la ligne de vie de l'émetteur et allant vers celle du cible.

Objet actif et passif

Un objet actif initie et contrôle le flux d'activité. Graphiquement, la ligne pointillée verticale d'un objet actif est remplacée par un double trait vertical.

Un objet passif a besoin qu'on lui donne le flux d'activité pour pouvoir exécuter un méthode.

Message synchrone

Exemple : l'invocation d'une opération.

L'émetteur reste bloqué le temps que dure l'opération, il attend le résultat de l'opération (un message).

Graphiquement, un message synchrone se représente par une flèche en traits pleins et à l'extrémité pleine partant de la ligne de vie de l'émetteur et allant vers celle du cible. Ce message peut être suivi d'une réponse qui se représente par une flèche en pointillé.

Message de création et destruction

La création d'une nouvelle instance d'une classe (un objet) est matérialisée par une flèche qui pointe sur le sommet d'une ligne de vie.

La desctruction d'un objet est matérialisée par une croix qui marque la fin de la ligne de vie de l'objet.

Condition, itération, envoie en parallèle, etc.

Il existe deux notations alternatives, selon la version de UML.