Cours 4 - C++ pour les mathématiques appliquées

Analyse et Conception Objet de Logiciels

La dernière fois . . .

- Constructeur(s)
- Opérateur(s)

Opérateur

Différence entre opérateur et constructeur!

Destructeur

Quand sont appelés les destructeurs!

const

virologie!

... aujourd'hui

- ACVL
- Diagrammes d'analyse

Introduction

Contexte

Problème du développement logiciel

> - dépassement budget et délais
> - cahier des charges non respectés

Dans les prochaines séances : analyse et conception

- Comment communiquer en amont du développement
- Comment concevoir les logiciels (approche objet)
- Comment comprendre et modifier du code existant

Analyse et Conception

Objectifs

- détailler les activités d'analyse et conception de logiciel
- pour des projets de taille *réelle*
- avec des approches objets
- en utilisant UML

Etude de cas

- système d'accès badgé à un bâtiment
- identification biométrique
- résolution d'équations
- reconstruction d'objets à partir d'un nuage de points

Analyse

Phase d'analyse

Objectifs

- comprendre les besoins du client
- effectuer une étude de faisabilité
- obtenir une bonne compréhension du domaine
- désambiguïser le cahier des charges
- obtenir une première ébauche de la structure du système

Expression des besoins

Principe

- comprendre les besoins du client
- exprimer ces besoins sous une forme compréhensible par le client et l'équipe de développement
- supporter/anticiper les évolutions

Problèmes liés aux besoins

- $1/3$ des problèmes rencontrés
- informations erronées
- cahier des charges incomplet
- évolutions incontrôlées

Point Clé Impliquer les utilisateurs!

Quels besoins ?

Types de besoins

- **Besoins fonctionnels :** fonctionnalités du logiciel
- **Fiabilité :** robustesse, couverture de tests, possibilité de récupération après une panne
- **Facilité d'utilisation :** ergonomie, aide, documentation
- **Rendement :** utilisation du CPU, temps de réponse
- **Portabilité :** plates-formes sur lesquelles le système doit pouvoir fonctionner
- **Maintenabilité :** effort nécessaire pour modifier le logiciel, correction, améliorations, adaptations

Analyse des besoins : comment ?

Une "analyse de texte" du cahier des charges

Aspects fonctionnels : verbes utilisés dans le cahier des charges.

Problèmes de la langue naturelle

- certains mots ne sont pas pertinents
- synonymes
- ambiguïtés
- beaucoup d'informations sont implicites (par exemple, on suppose souvent que le lecteur a une bonne connaissance du domaine)

Analyse de texte

But de l'analyse de texte

- préciser le cahier des charges
- lever certaines ambiguïtés

Insuffisance de l'analyse de texte

Le cahier des charges reste souvent insuffisant, imprécis, ambigu, incomplet, contradictoire.

Exemples industriels de cahier des charges

- une phrase dans un email
- des milliers de lignes dans un fichier excel

Produits de l'analyse des besoins

Documents issus de l'analyse des besoins

- Cahier des charges
  - description de l'environnement du système
  - reel du système
- Glossaire
- Manuel utilisateur
- Document de spécification globale

Expression des besoins fonctionnels

Spécification globale

- quelles fonctionnalités ?
- comment doit réagir le système du point de vue de l'utilisateur ?
- *on ne parle pas de comment ce sera fait* (pas d'architecture, conception détaillée implémentation)

Diagrammes

Diagrammes utilisés

1. **cas d'utilisations** et **diagramme de cas d'utilisation** (pour décrire les principales fonctionnalités)
2. **diagrammes de séquences** (pour illustrer des scénarios type)
3. **diagrammes états-transitions** (pour documenter le comportement du système du point de vue de l'utilisateur)

Diagramme des cas d'utilisations

Cas d'utilisation

Objectifs

- compréhension et reformulation du cahier des charges
- *structuration* des besoins fonctionnels

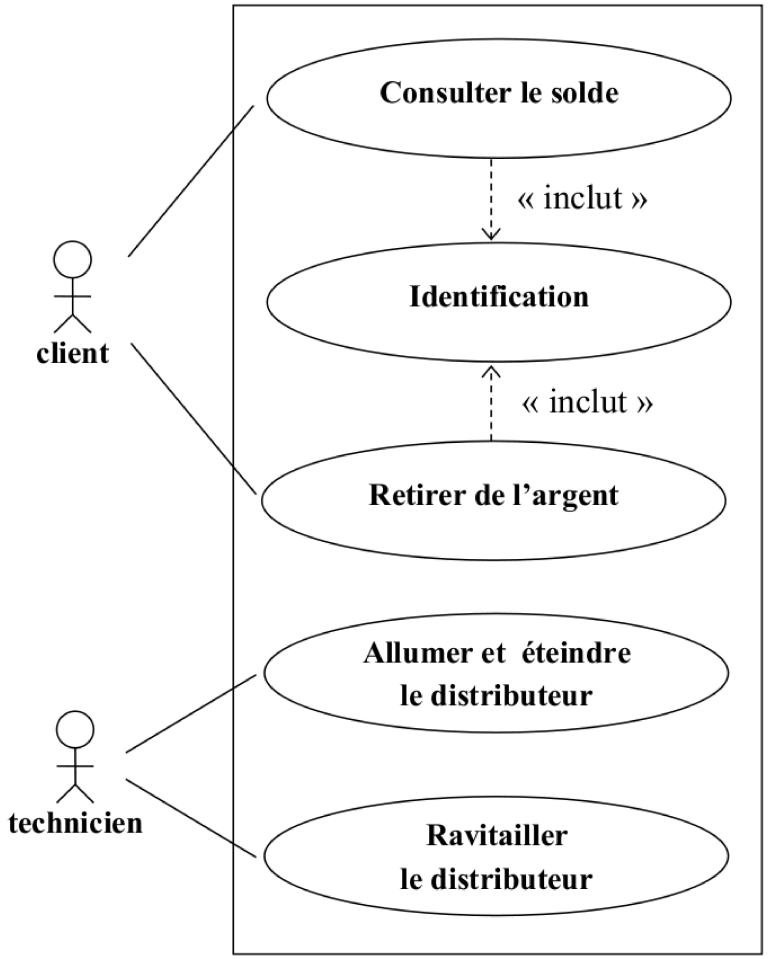
1. Acteur quelqu'un ou quelque chose qui interagit avec le système

- *acteurs principaux* : personnes qui utilisent les fonctionnalités principales du système
- *acteurs secondaires* : personnes qui effectuent des tâches de maintenance ou d'administration
- *le matériel externe ou autre système*

2. Un cas d'utilisation

- décrit une manière spécifique d'utiliser le système
- regroupe une famille de scénarios d'utilisation
- spécifie le comportement du système
- compréhensible (langue naturelle, terminologie du domaine)

Diagramme des cas d'utilisations



Granularité du cas d'utilisation

- un cas d'utilisation = une famille de scénarios d'utilisation
- pas de consignes pour la granularité… mais le diagrammes doit rester lisible…
- l'ensemble des cas d'utilisation doit couvrir l'ensemble des besoins
- l'énumération des cas d'utilisation doit être *structurée* souvent arborescente

Cas d'Utilisation : Description

Contient :

- nom
- pré-condition
- description : *interactions entre le système et les acteurs*
- post-condition
- exceptions : ce qui doit ce qui doit se passer si une pré-condition n'est pas satisfaite

Cas d'Utilisation : Description

S'écrit :

- en langue naturelle
- le plus précisément possible
- de façon *structurée*
- en précisant les interactions acteurs-système

Exemple de Description

- éviter "On"
- éviter les termes flous en général (glossaire)
- Structurer la description

Description structurée pour "Identification"

- 1 Le système affiche "entrer code"
    - 1.1 L'utilisateur entre son code et termine en appuyant sur "valider"
    - 1.2 Le système vérifie que le code est correct
        - 1.2.1 Si le code est correct ...
        - 1.2.2 Si le code est incorrect ...
- 2 ...

Structuration des Fonctionnalités du Logiciel

Les cas d'utilisation structurent les fonctionnalités.

Cette structuration peut être utilisée dans l'ensemble du cycle de vie du logiciel :

- spécification
- conception
- mise en oeuvre
- validation

Exemples

- Planification des versions successives
- Test

Mise en pratique

Exercice Gestion de l'ouverture/fermeture des portes, accès par carte

- concerne élèves, enseignants, chercheurs, administratifs; chacun a des droits d'accès différents au bâtiment.
- on utilise sa carte quand la porte est fermée. Si on a droit d'accès à cette heure, la porte s'ouvre
- à tout moment, un incident peut bloquer les portes
- à tout moment, le réparateur de portes/l'administrateur du logiciel peuvent bloquer/débloquer les portes

1. Identifier les acteurs
2. Dessiner le diagramme de cas d'utilisation

Expression des besoins fonctionnels

Diagrammes utilisés

1. **cas d'utilisations** et **diagramme de cas d'utilisation**(pour décrire les principales fonctionnalités)
2. **diagrammes de séquences** (pour illustrer des scénarios type)
3. **diagrammes états-transitions** (pour documenter le comportement du système du point de vue de l'utilisateur)

Diagrammes de séquence

Documentation des cas d'utilisation

- à un cas d'utilisation correspondent un ou plusieurs diagrammes de séquence,
- qui illustrent un/plusieurs scénarios d'utilisation du système.

Diagrammes de séquence

Diagrammes de séquence

Diagrammes de séquence système

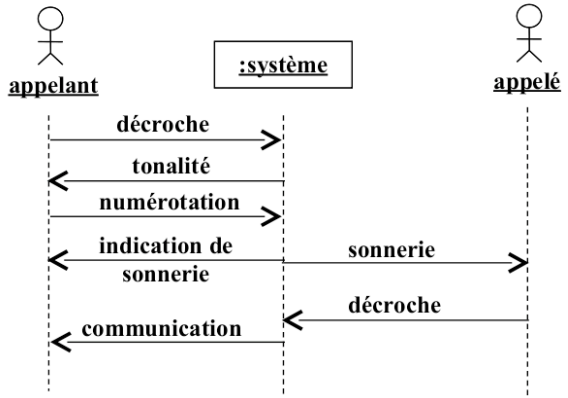
- système vu comme une boite noire
- acteurs impliqués dans le cas d'utilisation
- accent sur les interactions acteurs/système

Définition des évènements :

- auxquels le système peut réagir
- que le système peut générer

Diagramme de séquence système - Exemple

"Communication standard entre un appelant et un appelé"



Pré-conditions

- la numérotation correspond à un numéro correct
- l'appelé n'est pas déjà en communication
- l'appelé décroche le téléphone

Diagrammes de séquence

Utilisation des diagrammes de séquence pour illustrer les scénarios d'utilisation :

- Apprendre à lire et utiliser ces diagrammes
- Dans le cadre de l'analyse, on se limite à l'illustration de scénarios non clairement défini dans le cahier des charges (4 max.)
- Attention: conserver un diagramme lisible qui illustre un seul point !

Diagrammes de séquence

Tout diagramme de séquence a

- Un titre
- Une ou plusieurs pré-condition(s)
- Éventuellement des post-conditions

Diagrammes d'états - Transitions

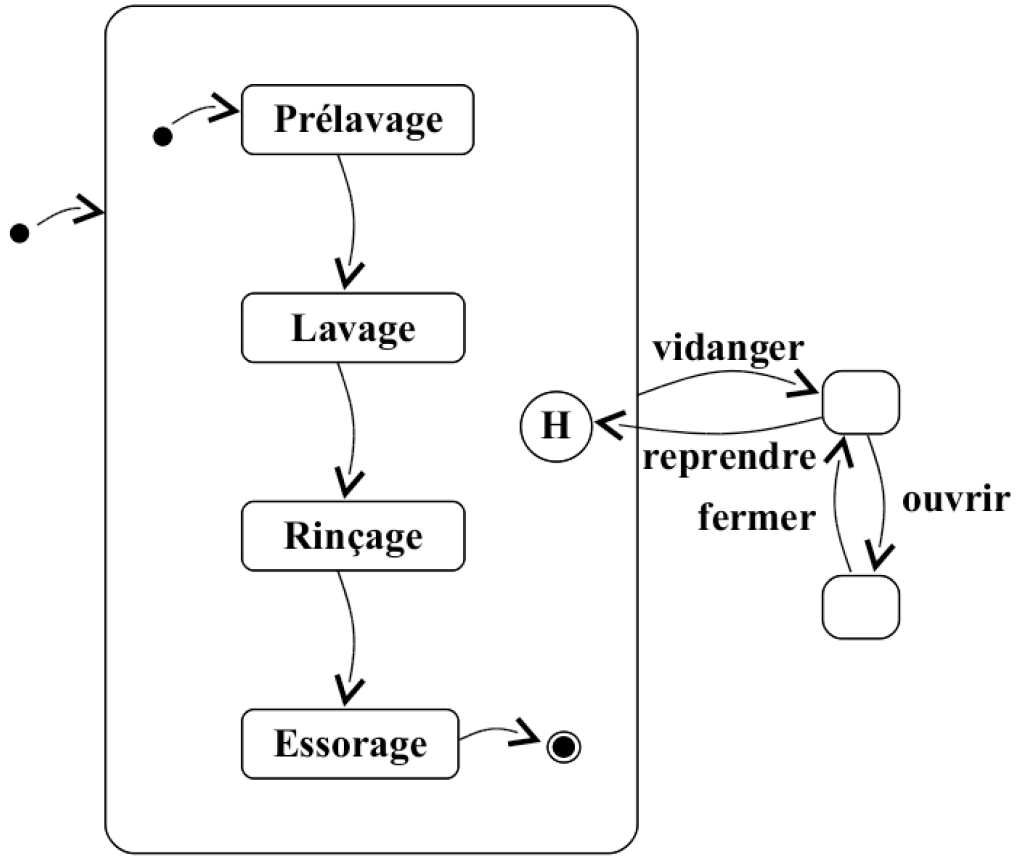
Diagrammes d'états - transitions

Spécifier le comportement général du système :

- états possibles du système
- évènements auxquels le système peut réagir (entrées)
- évènements que le système peut générer (sorties)

Diagrammes d'états - transitions : Exemple

Diagramme d'états-transitions d'une machine à laver



Exercice
Gestion de l'ouverture/fermeture des portes, accès par carte

- concerne élèves, enseignants, chercheurs, administratifs ; chacun a des droits d'accès différents au bâtiment
- on utilise sa carte quand la porte est fermée. Si on a droit d'accès à cette heure, la porte s'ouvre
- à tout moment, un incident peut bloquer les portes
- à tout moment le réparateur de portes/l'administrateur du logiciel peuvent bloquer/débloquer les portes
- Reprendre le diagramme de cas d'utilisation : dériver 4 diagrammes de séquence les plus pertinents
  Ne pas oublier les pré-conditions (et post-conditions si besoin).

Diagramme de classes d'analyse

Diagramme de classes d'analyse

Objectif : élaborer un "modèle objet" du système

Diagramme de classes d'analyse :

- Objets du monde réel
- Aspects statiques
- Structuration classes/relations

Diagramme de classes d'analyse

Utilisation d'UML :

- Diagrammes de classes
- "classes conceptuelles"

> Compréhensible par les utilisateurs
> Utilise la terminologie du domaine.

Diagramme de classes d'analyse {#diagramme-de-classes-danalyse-2}

Etapes de création du diagramme de classes d'analyse :

1. identifier les classes conceptuelles
2. identifier les associations entre ces classes, avec leurs multiplicités
3. identifier les attributs de chaque classe
4. (une classe conceptuelle ne comporte, en général, pas de méthodes)

Classes conceptuelles

Technique :

- on repère les noms et les groupes nominaux dans le cahier des charges
- ces noms permettent de déterminer les objets du monde réel qui doivent être représentés dans le système
- on regroupe ces objets en différentes classes

Relations entre classes conceptuelles

Identifier les relations entre les classes

1. agrégations, compositions
2. héritage
    - hiérarchies de classes par généralisation
    - factorisation des éléments communs à des objets de différentes classes (attributs, méthodes, relations)

Attributs de classes conceptuelles

Choix: Attribut vs. "classe + relation "

- **attribut :** type "primitif" (entier, booléen, chaîne de caractères...)
- **classe + relation :**
  - valeur composite
  - opérations associées aux valeurs
- abstraction généralisant plusieurs types

Diagramme de classes d'analyse : objectifs

Obtenir une première ébauche de la *structure du système* en

- obtenant une bonne compréhension du domaine
- éliminant des ambiguïtés du cahier des charges

Diagramme de classe : Utilisation

- Construction d'**un diagramme de classes "conceptuelles"**
- A un niveau "**spécification**", avant la conception
- On pourra ensuite s'inspirer du modèle objet du domaine pour
    - définir l'architecture du système
    - en faire la conception détaillée

Mise en pratique

Exercice Gestion de l'ouverture/fermeture des portes, accès par carte

- concerne les élèves, enseignants, chercheurs, administratifs ;\ chacun a des droits d'accès différents au bâtiment
- on utilise sa carte quand la porte est fermée. Si on a droit d'accès à cette heure, la porte s'ouvre
- à tout moment, une détection d'incendie peut bloquer les portes
- à tout moment le réparateur de portes/l'administrateur du logiciel peuvent bloquer/débloquer les portes

A retenir

- Diagramme des cas d'utilisations.
- Diagramme de séquence
- Diagramme état-transition

La prochaine fois

- Conception