



Analyse et Conception Objet de Logiciels

Chapitre 3 **Analyse**

Cycle de vie (rappel)

- Cycle de développement en cascade



**Recueillir
les exigences**

Analyser



Concevoir



Coder



**Intégrer, tester & effectuer
le contrôle qualité**



Analyse : objectifs

- Les objectifs de l'analyse sont les suivants :
 - ☐ comprendre les besoins du client
 - ☐ effectuer une étude de faisabilité
 - ☐ obtenir une bonne compréhension du domaine
 - ☐ éliminer le maximum d'ambiguïtés du cahier des charges
 - ☐ obtenir une première ébauche de la structure du système
- Expression des besoins :
 - ☐ comprendre et reformuler les besoins des utilisateurs
- Diagramme de classes d'analyse :
 - ☐ mettre en évidence les concepts significatifs du système



Expression des besoins

■ Gestion des besoins

- ☐ comprendre, exprimer et mémoriser les besoins du système sous une forme compréhensible par le client et l'équipe de développement.
- ☐ Les besoins ne sont pas figés une fois pour toutes au début du projet, mais sont amenés à évoluer, en particulier si on utilise un modèle de cycle de vie itératif.

■ Contrôler la gestion et l'évolution des besoins

- ☐ Important, car un tiers des problèmes rencontrés lors du développement de logiciels viennent de cette étape

■ Pourquoi ?

- ☐ les utilisateurs peuvent fournir des informations erronées ;
- ☐ les besoins du système peuvent être incomplets ;
- ☐ les besoins des utilisateurs évoluent d'une façon incontrôlée.



Types des besoins

- On distingue :
 - les besoins fonctionnels, qui concernent les fonctionnalités du logiciel,
 - les besoins non fonctionnels.

- Parmi les besoins non fonctionnels, on trouve :
 - la fiabilité (robustesse, possibilité de récupération après une panne) ;
 - la facilité d'utilisation (ergonomie, aide, documentation) ;
 - efficacité (temps de réponse) ;
 - portabilité ;
 - maintenabilité (facilité à corriger des erreurs, à faire des améliorations ou des adaptations du logiciel) ;
 - effort de validation (tests, couverture des tests).



Analyse des besoins

- Analyse du cahier des charges :
 - ☐ préciser le cahier des charges et de lever certaines ambiguïtés.
 - ☐ il peut être utile de rédiger un glossaire qui définit les principaux termes utilisés.

- Le cahier des charges est non seulement imprécis et ambigu, mais également incomplet, voire contradictoire.
 - ☐ organiser des entrevues avec les utilisateurs, les techniciens, les gestionnaires ;
 - ☐ demander aux utilisateurs de remplir des questionnaires ;
 - ☐ on peut observer les activités des utilisateurs sur site, afin de mieux comprendre leurs besoins.



Analyse des besoins

- A l'issue de l'analyse des besoins, les documents suivants peuvent être produits :
 - un cahier des charges, qui comporte :
 - une description de l'environnement du système : les machines, le réseau, les périphériques (imprimantes, capteurs. . .), l'environnement physique. . . etc.
 - le rôle du système ;
 - un glossaire, qui définit les principaux termes utilisés dans le cahier des charges ;
 - un manuel utilisateur ;
 - un document de spécification globale, précisant les besoins fonctionnels et non fonctionnels du système.



Analyse des besoins

- Expression des besoins fonctionnels
 - cas d'utilisation ;
 - diagrammes de séquence ;
 - diagrammes d'états-transitions.



1. Expression des besoins fonctionnels

Rédaction de cas d'utilisation

- Objectif : reformuler les besoins fonctionnels du système.
 - comprendre et clarifier le cahier des charges, à travers une reformulation ;
 - structurer les besoins.
- Un cas d'utilisation décrit ce que le système doit faire, du point de vue des utilisateurs, sans décrire comment cela sera implémenté.
- Un cas d'utilisation :
 - décrit une manière spécifique d'utiliser le système ;
 - regroupe une famille scénarios d'utilisation du système.



1. Expression des besoins fonctionnels

Rédaction de cas d'utilisation

- On utilise le langage naturel et on utilise la terminologie des utilisateurs.
- Les cas d'utilisations doivent être compréhensibles à la fois par les utilisateurs et les développeurs.
- Un cas d'utilisation regroupe une famille de scénarios d'utilisation suivant des critères fonctionnels.
- C'est une abstraction du dialogue entre les acteurs et le système.
- Les scénarios seront décrits ensuite par d'autres diagrammes (diagrammes d'interaction, diagrammes d'états-transitions).



1. Expression des besoins fonctionnels

Rédaction de cas d'utilisation

- Pour exprimer ces besoins fonctionnels, on commence par définir les acteurs.
- Un acteur est quelqu'un ou quelque chose qui interagit avec le système (personne, environnement, autre système).
- La détermination des acteurs permet de préciser les limites du système.
- Lorsqu'il y a beaucoup d'acteurs, on les regroupe par catégories :
 - les acteurs *principaux* : personnes qui utilisent les fonctions principales du système ;
 - les acteurs *secondaires* : personnes qui effectuent des tâches administratives ou de maintenance
 - le matériel externe : les dispositifs matériels périphériques ;
 - les autres systèmes (avec lesquels le système interagit).



1. Expression des besoins fonctionnels

Rédaction de cas d'utilisation

- Chaque cas d'utilisation peut comprendre :
 - **une pré-condition** (condition qui doit être satisfaite pour que la fonctionnalité puisse être utilisée) ;
 - **une description** de la suite des interactions entre le système et les acteurs, en distinguant les différents scénarios possibles.
 - **une post-condition** (condition satisfaite une fois l'interaction terminée) ;
 - **des exceptions** (cas exceptionnels, ou ce qui se passe si une pré-condition n'est pas satisfaite).



1. Expression des besoins fonctionnels

Rédaction de cas d'utilisation

- Description de cas d'utilisation :

- 1. Le système demande à l'utilisateur d'entrer son code secret.

- 1.1. L'utilisateur entre son code secret et termine en appuyant sur « valider ».

- 1.2. Le système vérifie que le code secret entré par l'utilisateur est correct.

- 1.2.1. Si le code est correct...

- 1.2.2. Si le code est incorrect...

- 2. ...

- Cas d'utilisation : structurent les différentes fonctionnalités du logiciel
- Structuration utilisée dans l'ensemble du développement du logiciel, de la spécification aux tests



1. Expression des besoins fonctionnels

Scénarios de haut niveau

- Diagrammes de séquence : illustrent les différents cas d'utilisations
- Chaque cas d'utilisation est accompagné de plusieurs diagrammes de séquence qui montrent différents scénarios de fonctionnement du système logiciel.

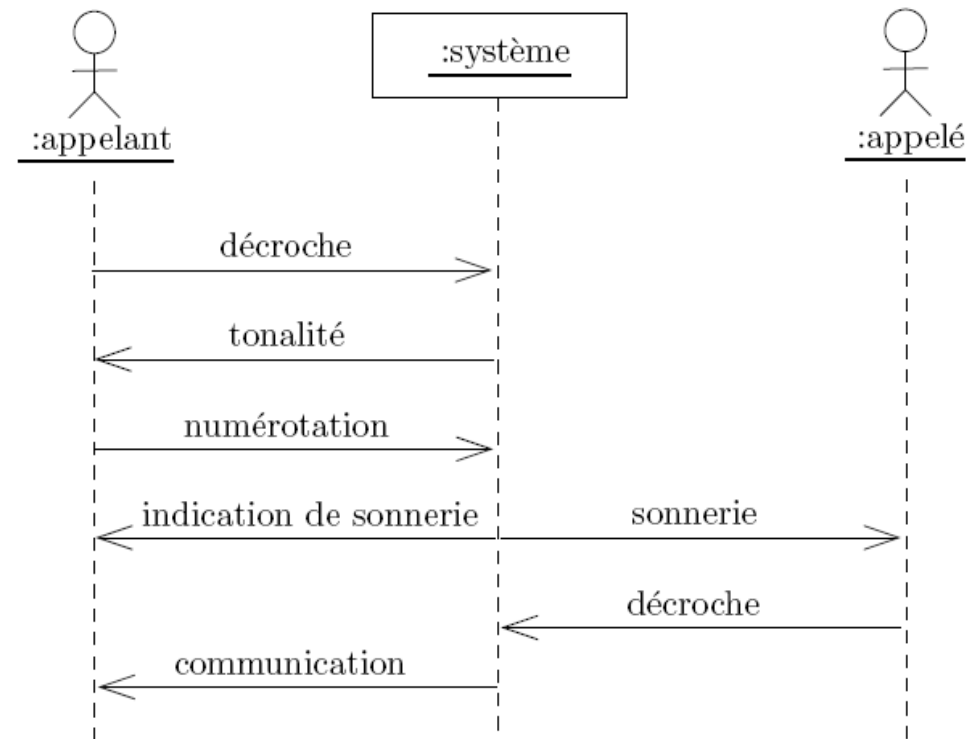
1. Expression des besoins fonctionnels

Scénarios de haut niveau

« Diagrammes de séquence système »

- Ce scénario peut se dérouler sous un certain nombre d'hypothèses, ou pré-conditions :

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





1. Expression des besoins fonctionnels

Diagrammes d'états-transitions

- Les diagrammes d'états-transitions permettent de spécifier le comportement général du système, en précisant :
 - les états possibles du système ;
 - tous les enchaînements possibles d'opérations.
- « Diagrammes d'états-transitions de protocole » :
 - diagrammes dans lesquels on ne représente que les événements auxquels le système peut réagir.
- On ne représente pas les actions internes au système.
- Ces diagrammes permettent de définir l'ensemble des séquences d'événements intéressantes pour utiliser une fonctionnalité du système.



2. Diagramme de classes d'analyse

Objectif

Identifier les éléments du monde réel qui sont utiles pour le système, et de faire abstraction des détails inutiles.

- On élabore un ou plusieurs diagrammes de classes qui représentent les classes significatives du système
- Ces diagrammes de classes sont constitués de classes dans lesquelles aucune opération n'est définie.
- On représente :
 - les classes conceptuelles ;
 - les attributs de ces classes ;
 - les associations entre ces classes.



2. Diagramme de classes d'analyse

Classes conceptuelles

- Une classe conceptuelle comporte :
 - un *nom* ;
 - une *intension* (qui indique ce que représente cette classe) ;
 - une *extension* (l'ensemble des objets instances de cette classe, à un instant donné).
- Identification des classes conceptuelles
 - Repérer les noms et groupes nominaux dans le cahier des charges
 - Identifier :
 - synonymes : deux mots représentent le même concept
 - ambiguïtés : un mot représente deux concepts différents



2. Diagramme de classes d'analyse

Relations entre les classes conceptuelles

- Relations « simples »
 - Eviter de faire apparaître les relations *dérivées*
- Agrégations et compositions
 - Un élément « appartient » à, ou « est contenu » dans un autre élément
- Généralisations
 - Factoriser des attributs ou associations communs à plusieurs classes



2. Diagramme de classes d'analyse

Attributs

- Les attributs doivent être de types primitifs (entier, réel, booléen, chaîne de caractères...)
- Valeurs de type non primitif :
 - On introduit une nouvelle classe et une association.



Exercise 14



Description du cas d'utilisation « minuterie ».

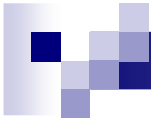
- 1. La minuterie est composée de trois compteurs (compteur des heures, compteur des minutes et compteur des secondes), de quatre boutons (Mode, Incr, Start/Stop et Fermer) et d'un indicateur d'alarme.
- 2. La minuterie peut être en marche ou arrêtée.
 - 2.1 Minuterie arrêtée. Si la minuterie est arrêtée, elle peut être dans trois modes : le mode « normal », le mode « édition heures », le mode « édition minutes ». Si les trois compteurs ne sont pas à zéro, l'utilisateur peut démarrer la minuterie en appuyant sur Start/Stop.
 - 2.1.1 Mode « normal ». L'utilisateur peut passer au mode modification heures en appuyant sur Mode.
 - 2.1.2 Mode « édition heures ». L'utilisateur peut incrémenter les heures en appuyant sur Incr. Si le compteur des heures est égal à 23, il repasse à 0. L'utilisateur peut passer au mode « édition minutes » en appuyant sur Mode.
 - 2.1.3 Mode « édition minutes ». L'utilisateur peut incrémenter les minutes en appuyant sur Incr. Si le compteur des minutes est égal à 59, il repasse à 0. L'utilisateur peut revenir au mode normal en appuyant sur Mode.



- 3. Minuterie en marche.
 - 3.1 La minuterie décrémente son compteur de secondes chaque seconde.
 - 3.1.1 Si le compteur de secondes est différent de 0, il est décrétementé d'une unité.
 - 3.1.2 Si le compteur de secondes est égal à 0 et le compteur de minutes est différents de 0, le compteur de minutes est décrétementé et le compteur de secondes passe à 59.
 - 3.1.3 Si les compteurs de secondes et de minutes sont à 0 et le compteur d'heures est différent de zéro, les compteurs de secondes et de minutes passent à 59 et le compteur d'heures est décrétementé.
 - 3.1.4 Si les trois compteurs sont à 0, l'alarme est activée et la minuterie s'arrête.
 - 3.2 L'utilisateur peut arrêter la minuterie en appuyant sur Start/Stop.



- 4. L'alarme peut être active ou inactive.
 - 4.1 Alarme inactive.
 - 4.1.1 Lorsque la minuterie est en marche, l'alarme est inactive.
 - 4.1.2 L'alarme est activée lorsque la minuterie est en marche et qu'elle atteint 0.
 - 4.2 Alarme active.
 - 4.2.1 Niveaux d'alarme. Lorsqu'une alarme est active, elle peut être au niveau 1, 2 ou 3. Lorsqu'une alarme est activée, elle est au niveau 1. Elle passe ensuite au niveau 2 après une minute, puis au niveau 3, après deux minutes, puis est désactivée après trois minutes.
 - 4.2.2 Désactivation de l'alarme. Lorsqu'une alarme est active, l'utilisateur peut la désactiver en appuyant sur le bouton Start/Stop. Après une minute en niveau trois, l'alarme est désactivée



- 5. Réactions au temps qui passe.
 - 5.1 Le système réagit de deux façons au temps qui passe.
 - 5.1.1 Lorsque la minuterie est en marche, les compteurs sont décrémentés d'une seconde à chaque seconde qui s'écoule.
 - 5.1.2 Lorsque l'alarme est active, elle change de niveau toutes les minutes.
 - 5.2 Ces deux réactions sont indépendantes car l'alarme ne peut être active lorsque la minuterie est en marche.