

Cas d'utilisation, User Stories

Diagrammes de cas d'utilisation UML

Des outils pour le recueil des exigences fonctionnelles

HAI401I
Modélisation et Programmation par Objets 2

Université de Montpellier

Plan

- 1 Introduction
- 2 Cas d'utilisation
- 3 Diagrammes UML
- 4 User stories
- 5 Synthèse

Ingénierie des exigences (requirements)

Phase amont de la conception des systèmes

Défi

« Réduire la fréquence et l'amplitude des problèmes de développement de systèmes en appliquant en amont des méthodes, techniques et outils qui garantissent (par leur dimension systématique) une meilleure prise en compte des exigences dans le développement des systèmes ^a ».

a. C. Salinesi, Univ. Paris Panthéon – Sorbonne, HDR, 2010

Ingénierie des exigences (requirements)

Développement motivé dans les années 90 par un constat réalisé par le Standish Group (<https://www.standishgroup.com/>) sur l'échec de nombreux projets informatiques

Principaux facteurs de succès (Chaos Report années 90)

- User involvement
- Executive management support
- **Clear Statement of Requirements**
- Proper planning
- Realistic expectations

Les facteurs de succès ont beaucoup évolué depuis (le Standish Group publie un rapport tous les 2 ans).

Ingénierie des exigences (requirements)

Standish Group Chaos Report (2020) Beyond Infinity

« Incomplete requirements cause challenged and failed projects »
serait un mythe

Que doit-on en penser ?

Depuis 10 ans, les rapports montrent la supériorité des projets développés :

- avec des méthodes **AGILE** ou proches comme SCRUM
comparativement aux projets développés de manière traditionnelle
- dans des équipes disposant d'un environnement **humain** propice
(« good sponsor, good team and good place »)

Développer l'analyse des exigences

Doit se faire de manière AGILE et continue (petites étapes, cycles courts, relation proche avec l'utilisateur)

<https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/>

<https://hennypoortman.wordpress.com/2021/01/06/review-standish-group-chaos-2020-beyond-infinity/>

Ingénierie des exigences (requirements)

Éléments entrant en jeu

- Déterminer ce que veut un client, répondre à un appel d'offre, rédiger un contrat
- Exigences fonctionnelles (services rendus) et non-fonctionnelles/extra-fonctionnelles (ex. performance, outils imposés, normes à respecter)
- De nombreux partenaires : client, utilisateurs, chefs de projet, experts, analystes, architectes, développeurs, testeurs, etc.
- Aspects : légaux, contractuels, économique, social

Quelques langages et méthodes

- *use cases* introduits en 1987 par I. Jacobson
- scripts et scénarios (J. Hooper)
- langages formels, comme les réseaux de Petri
- langages de modélisation de buts comme KAOS (A. Dardenne) ou I* (J. Mylopoulos)
- diagrammes de cas d'utilisation en UML (1995) et méthodologie décrite dans le Unified Process (1997)
- templates de description de use case par A. Cockburn (2001)
- récits utilisateurs (user stories), cartes de récits (K. Beck 1999, M. Cohn 2004)
- adaptation des use cases au contexte AGILE, SCRUM (Use Case 2.0, I. Jacobson et al. 2011)
- cartographie des récits (J. Patton, 2014)

Plan

- 1 Introduction
- 2 Cas d'utilisation**
- 3 Diagrammes UML
- 4 User stories
- 5 Synthèse

Cas d'utilisation (définition)

Use Case

« A use case is all the ways of using a system to achieve a particular goal for a particular user. Taken together the set of all the use cases gives you all of the useful ways to use the system, and illustrates the value that it will provide. » Ivar Jacobson, Ian Spence, Kurt Bittner.

<https://www.ivarjacobson.com/publications/white-papers/use-case-ebook>

Cas d'utilisation (A. Cockburn)

Use Case

- Contrat entre les partenaires d'un système à propos de son comportement
- Formats multiples, textuels et graphiques

Rôle

- Protéger les intérêts des partenaires
- Permettre la communication, la discussion, la négociation
- Support pour l'élaboration des tests

Use Case - Codification textuelle (A. Cockburn)

- Cas d'utilisation : ... Déclencheur : ...
- Acteur principal : *celui qui a un but précis*
- Portée : *Identifie le système dont on parle*
- Niveau : *Utilisateur (simple) ou Summary (complexe)*
- Partenaires et leurs préoccupations : ...
- Précondition : *à vérifier pour que le cas d'utilisation se passe bien*
- Garantie minimale : *propriété qui doit être vraie quel que soit le déroulé*
- Garantie en cas de succès : *propriété qui doit être vraie si tout s'est bien passé*
- Scénario principal (nominal) : ...
- Extensions : *problèmes sur le scénario nominal*
- Variations : *autres possibilités*

Use Case - Codification textuelle (A. Cockburn)

- Cas d'utilisation : *Emprunt dans une bibliothèque*
- Déclencheur : *Demande d'emprunt*
- Acteur principal : *Emprunteur (E)*
- Portée : *Système d'information de la bibliothèque*
- Niveau : *But utilisateur*
- Partenaires et leurs préoccupations :
 - *Emprunteur (E) : abonné qui veut emprunter un ouvrage*
 - *Bibliothèque (B) : veut connaître l'emprunt pour vérifier qu'il est possible et archiver le cas échéant*
 - *Stock (S) : donne des informations sur les ouvrages et gère le stock*
 - *Catalogue (C) : donne des informations sur la référence (son existence, le public visé)*

Use Case - Codification textuelle (A. Cockburn)

- Précondition : *L'emprunteur est un abonné inscrit dans le système*
- Garantie minimale : *L'emprunteur a une information sur l'existence de la référence et sur la possibilité qu'il a de l'emprunter ; des explications ; ou une promesse d'avoir cette information si le système est out*
- Garantie en cas de succès : *L'emprunteur repart avec un exemplaire de l'ouvrage. Les informations correspondantes sont archivées par la bibliothèque, le stock est mis à jour*

Use Case - Codification textuelle (A. Cockburn)

- Scénario principal (nominal) :

- ① *E demande à emprunter un ouvrage dont il donne l'ISBN*
- ② *B consulte dans C les informations relatives à cette référence, puis vérifie dans S qu'un exemplaire est disponible et que l'abonné peut emprunter :*
 - il a moins de 5 emprunts en cours
 - il est majeur ou bien il emprunte un ouvrage pour public junior ou enfant
- ③ *B récupère l'ouvrage par S et le donne à E. S déclare l'ouvrage emprunté et archive le nom de l'emprunteur*
- ④ *B met à jour la liste des emprunts de l'emprunteur*

Use Case - Codification textuelle (A. Cockburn)

- Extensions de l'étape 2 :
 - 2a *l'ouvrage n'est pas référencé* → l'emprunteur est informé
 - 2b *il n'y a pas d'exemplaire disponible* → l'emprunteur est informé des dates de retour des exemplaires
 - 2c *l'emprunteur a déjà 5 emprunts en cours (c'est le maximum)* → l'emprunteur est informé et on lui donne la liste des emprunts qu'il a en cours
 - 2d *l'emprunteur est mineur et l'ouvrage est pour un public majeur* → l'emprunteur est informé
 - 2e *le système est hors service* → l'emprunteur est informé des dates prévues de reprise

Use Case - Codification textuelle (A. Cockburn)

- Extensions de l'étape 3 :

- 3a *l'ouvrage n'est pas en rayon alors que S le connaît comme disponible*
→ l'emprunteur est informé et un processus d'enquête interne à B et S est lancé
- 3b *l'emprunteur est parti ou a changé d'avis* → l'ouvrage est remis dans le stock

- Extensions de l'étape 4 :

- 4a *le système B ou S tombe en panne* → l'emprunt est noté dans un cahier spécial pour être saisi plus tard

Use Case - Codification textuelle (A. Cockburn)

- Variations :

- 1a E ne connaît pas l'ISBN mais a une autre information (nom de l'auteur et titre) → une recherche est faite dans C pour connaître l'ISBN
- 1b E consulte librement C pour choisir un ouvrage.

Plan

- 1 Introduction
- 2 Cas d'utilisation
- 3 Diagrammes UML**
 - Diagrammes de cas d'utilisation
 - Diagrammes de séquence
- 4 User stories
- 5 Synthèse

Cas d'utilisation en UML

En UML, deux types de diagrammes principaux sont utilisés pour représenter les cas d'utilisation.

Diagramme de cas d'utilisation

- il donne une vue logique
- il organise les cas d'utilisation et les connecte aux acteurs

Diagramme de séquence

- il donne une vue chronologique
- il présente l'ordre d'appel de fonctionnalités dans un scénario (notamment le scénario nominal)

Ces diagrammes ne dispensent pas de rédiger des textes, mais ils peuvent donner une vue plus synthétique

Diagramme de cas d'utilisation

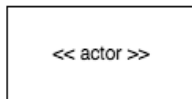
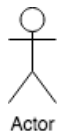
Objectif

- délimiter le système
- indiquer les acteurs
- indiquer les fonctionnalités externes
- préciser la logique d'inclusion

Diagramme de cas d'utilisation

Notation des 2 principaux concepts

- Acteur : entité extérieure au système ; acteur humain ou autre système



- Cas d'utilisation : toute manière d'utiliser le système

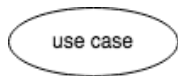
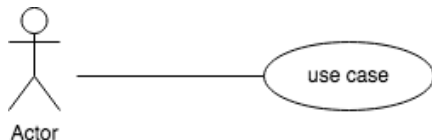


Diagramme de cas d'utilisation

Relations

- Participation (association) : un acteur participe à un cas



- Inclusion : A inclut obligatoirement B

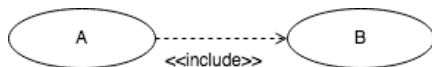


Diagramme de cas d'utilisation

Relations

- Extension : A inclut optionnellement B (B étend A ; les points d'extension peuvent être précisés)

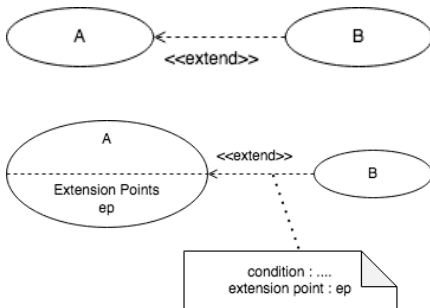


Diagramme de cas d'utilisation

Relations

- Spécialisation entre acteurs et entre cas

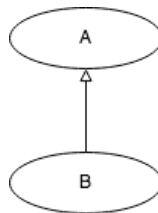


Diagramme de cas d'utilisation

Logiciel de gestion de bibliothèque

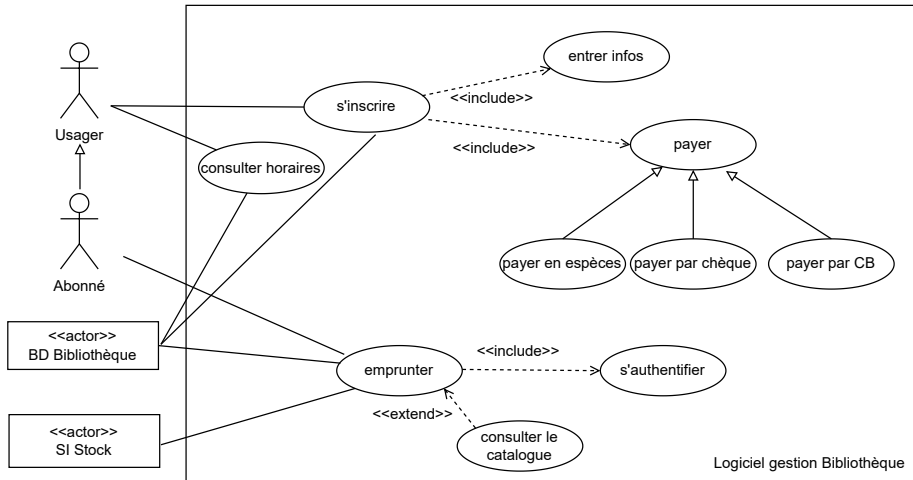


Diagramme de cas d'utilisation

Logiciel de gestion de bibliothèque

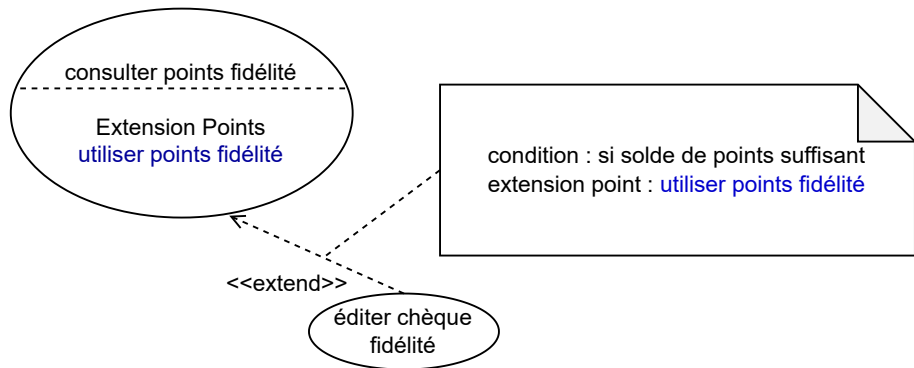


Diagramme de cas d'utilisation

Logiciel de gestion de bibliothèque

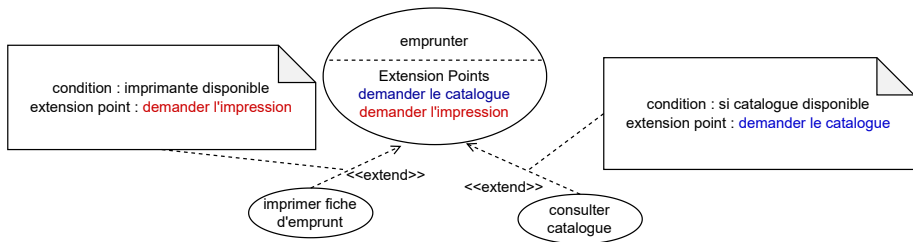


Diagramme de cas d'utilisation

Différencier extend et spécialisation entre cas : une question de point de vue et de description

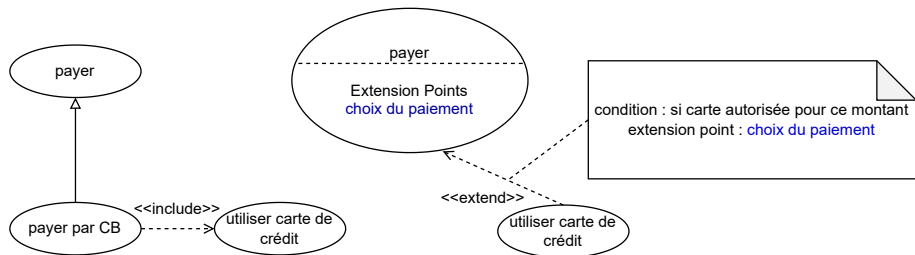
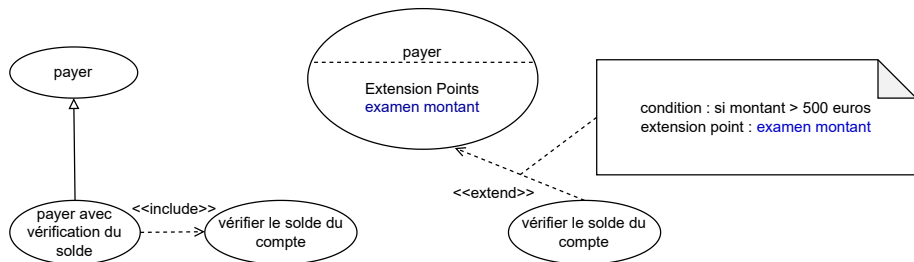


Diagramme de cas d'utilisation

Différencier extend et spécialisation entre cas : une question de point de vue et de description



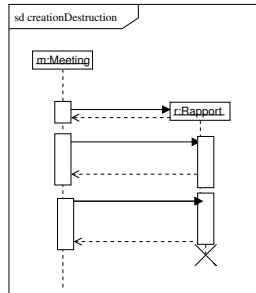
Diagrammes de séquence

L'un des diagrammes dynamiques proposés par UML

- Les diagrammes de séquence permettent de représenter les interactions entre des instances particulières. Un diagramme met en jeu :
 - des instances, et éventuellement des acteurs (dans le cadre d'un diagramme de cas d'utilisation),
 - des messages échangés par ces instances. Un message définit une communication entre instances. Ce peut être par exemple l'émission d'un signal, ou l'appel d'une opération.
- Le diagramme de séquence permet d'insister sur la **chronologie** des interactions : le temps s'écoule principalement du haut vers le bas.
- Le diagramme décrit un ensemble cohérent de messages, dans le cadre d'une **fonctionnalité du système**.

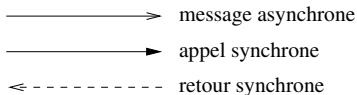
La ligne de vie

- À chaque instance est associée une ligne de vie, qui représente la vie de l'objet.
- Les événements survenant sur une ligne de vie (réception de message ou envoi de message) sont ordonnés chronologiquement.
- La ligne de vie est représentée par une ligne pointillée quand l'instance est inactive, et peut être représentée par une boîte blanche ou grisée quand l'instance est active.
- Quand une instance est détruite, on stoppe la ligne de vie par une croix.



Les messages

- Les messages sont représentés par des lignes fléchées.
- À chaque extrémité de la ligne fléchée correspond un événement (réception ou envoi).
- Le sens de la flèche permet de déterminer dans quel sens va le message.
- synchronie/asynchronie :



Syntaxe des noms de message

La syntaxe pour le nom d'un message est :

`([attribut =] signal-ou-NomOperation [([liste-arguments]]) [:
valeur-retour])`

`| *`

où la syntaxe pour un argument est :

`([nomParam =] valeur-argument) | (attribut = nomParamOut [:
valeurArgument]) | -`

* signifie : n'importe quel type de message

- signifie : paramètre indéfini

Par exemple, on peut avoir les noms de message suivants :

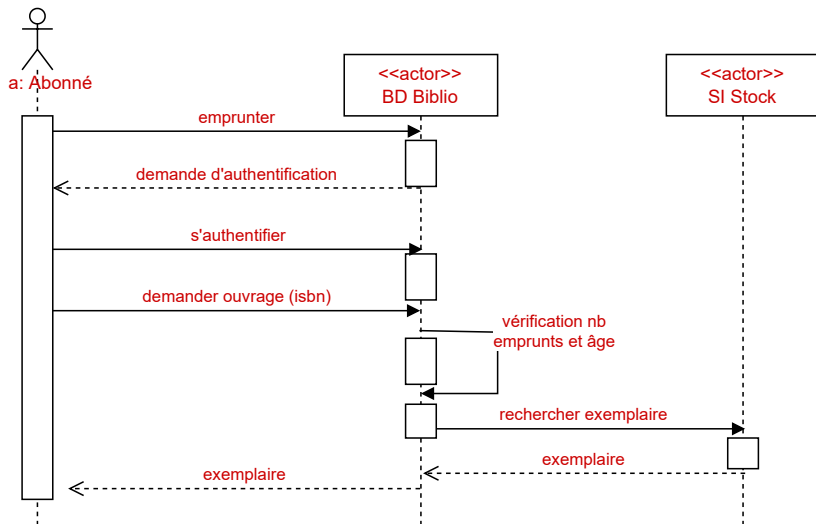
- `getAge()`
- `getAge() :12`
- `age=getAge() :12`
- `setAge(age=15)`
- `setAge(-)`

Pour aller plus loin ... fragments et autres opérateurs

- alternative
- composition parallèle
- itération

Diagramme de séquence

Séquence possible pour un emprunt entre acteur et système



Plan

- 1 Introduction
- 2 Cas d'utilisation
- 3 Diagrammes UML
- 4 User stories**
- 5 Synthèse

Récits utilisateurs

Principes

- Description simple d'une exigence
 - Un titre
 - Un niveau de priorité
 - Une estimation de sa complexité
 - Format "En tant que" (qui), "je souhaite" (quoi), "afin de" (pourquoi)
 - Critères INVEST
 - (qui) peut être lié à un personnage fictif représentant un utilisateur
 - Critères d'acceptation

Récits utilisateurs

Emprunt abonné ; priorité 1 ; moyen

- En tant que : abonné
- Je souhaite : emprunter un ouvrage
- Afin de : lire cet ouvrage

Vérif employé ; priorité 1 ; simple

- En tant que : bibliothécaire
- Je souhaite : vérifier le nombre d'emprunts d'un abonné
- Afin de : savoir s'il peut emprunter un exemplaire

Vérif stock ; priorité 1 ; simple

- En tant que : bibliothécaire
- Je souhaite : vérifier la présence d'un exemplaire
- Afin de : savoir si un ouvrage est disponible pour un prêt

Critères INVEST

Critères INVEST

- Indépendante
- Négociable
- Valorisable (avec de la valeur pour l'utilisateur)
- Estimable (pour la planification)
- Size (de petite taille)
- Testable

Critères d'acceptation

Critères d'acceptation menant aux tests

- Format "Etant donné que," "Lorsque," "Alors,"
- Exemple avec Vérif stock
 - En tant que : bibliothécaire
 - Je souhaite : vérifier la présence d'un exemplaire
 - Afin de : savoir si un ouvrage est disponible pour un prêt
- "Etant donné que," je suis sur la page de consultation du stock
- "Lorsque," j'entre un numéro ISBN
- "Alors," je reçois le nombre d'exemplaires de cette référence qui sont dans le stock

Plan

- 1 Introduction
- 2 Cas d'utilisation
- 3 Diagrammes UML
- 4 User stories
- 5 Synthèse**

Synthèse

- Récits utilisateurs

- + Simples à décrire, à lire et à mettre en œuvre (une itération de développement AGILE), aident à discuter oralement avec les utilisateurs, peut être écrit par l'utilisateur, sert à estimer le travail
- Manque de précision, de structure et de contexte (pas de vue globale), difficile de se rendre compte de la complétude, manque de temporalité

- Cas d'utilisation et diagrammes

- + Niveau de structure et de détails importants et précis, peuvent s'adapter au contexte AGILE à condition de les décomposer et de les faire évoluer, vision globale, spécification semi-formelle
- Plus complexes à écrire (réalisés par un analyste), à lire et à modifier en cas de changement

- Les deux approches peuvent être utilisées en synergie : réaliser des cas d'utilisation et les découper en récits ou l'inverse (voir [https ://martinfowler.com/bliki/UseCasesAndStories.html](https://martinfowler.com/bliki/UseCasesAndStories.html))