

Analyse et Conception Orientée Objet

Analyse Orientée Objet

A est un bon modèle de B si A permet de répondre de façon satisfaisante à des questions prédéfinies sur B.

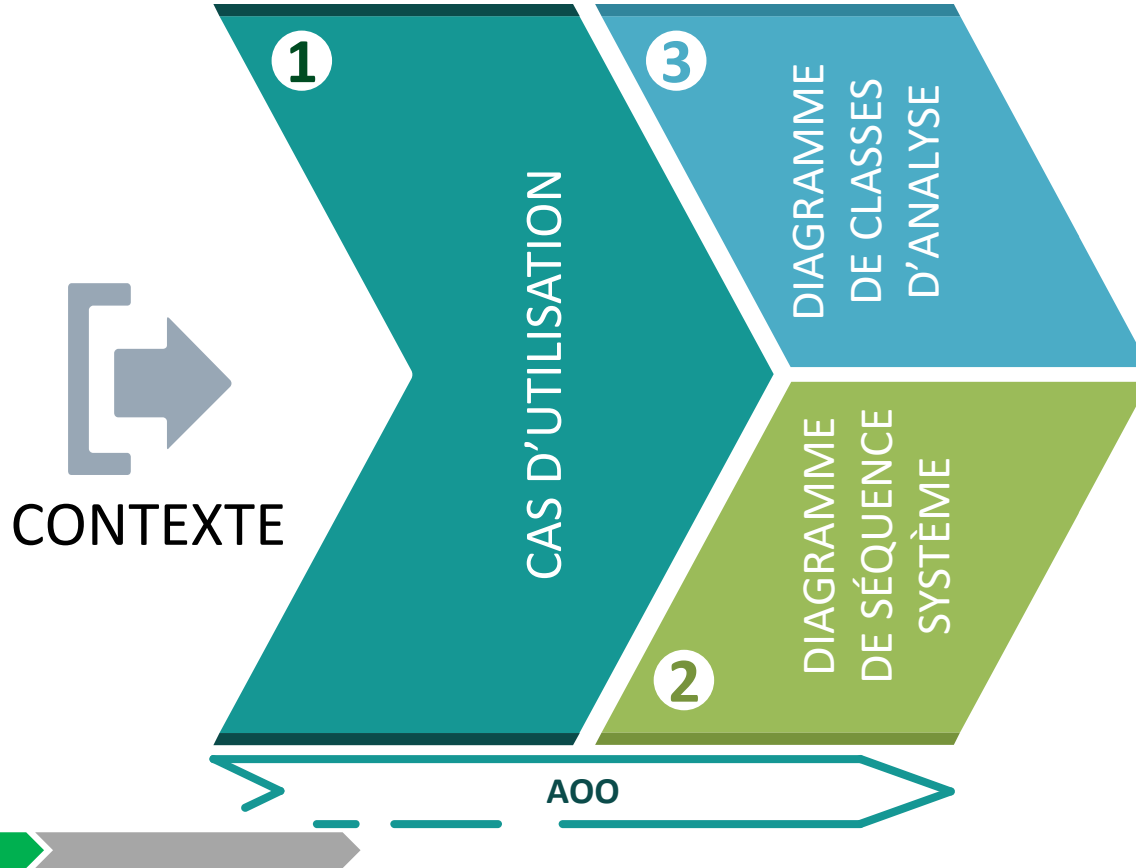
Douglas. T. Ross

Plan

- Pourquoi l'Analyse Orientée Objet (AOO)?
- **Étape 1:** Réalisation de cas d'utilisation
- **Étape 2:** Diagramme de séquence "système"
- **Étape 3:** Diagramme de classes d'analyse



Analyse Orientée Objet (AOO)



Plan

- Analyse Orientée Objet (AOO)
- Étape 1: Réalisation de cas d'utilisation
- Étape 2: Diagramme de séquence “système”
- Étape 3: Diagramme de classes d'analyse



Analyse – Etat de l'art

«L'analyse consiste à valider que le système qu'on va concevoir répond à un besoin bien identifié. Sachant qu'un besoin peut être exprimé ou implicite, avoué ou inavoué, latent ou potentiel...»



Comment le client l'a souhaité



Comment le chef de projet l'a compris



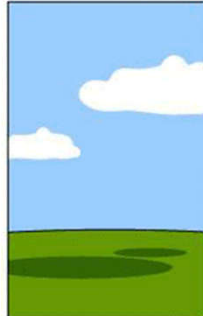
Comment l'analyste l'a schématisé



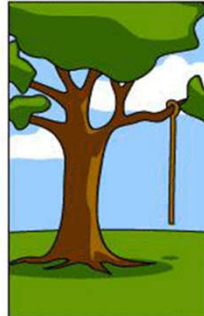
Comment le programmeur l'a écrit



Comment le Business Consultant l'a décrit



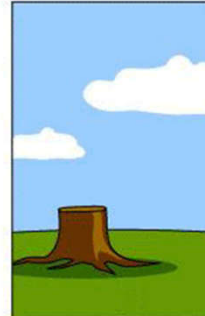
Comment le projet a été documenté



Ce qui a été installé chez le client



Comment le client a été facturé



Comment le support technique est effectué



Ce dont le client avait réellement besoin

Analyse et analyse

Analyse du domaine - analyse du métier

- Concepts : objets métier, objets du domaine.
- L'analyse du domaine peut être une tâche de fond transversale aux projets.

Analyse système

- Concepts : Objets métier pertinents pour le système à l'étude.
- L'analyse système concerne un système ou application particulière.

UML Analyse

- **Dans le cadre de ce cours lors de la phase d'analyse nous utiliserons trois (3) types de diagrammes :**
 - Cas d'utilisation
 - Diagramme de séquence
 - Diagramme de classes
- **Il est possible d'utiliser d'autres diagrammes UML lors de la phase d'analyse par contre il faut être en mesure de le justifier.**
- **20% de la notation UML est utilisée 80% du temps.**

Les idées clés de l'A00

- Guidée par les cas d'utilisation
- Pilotée de façon itérative et incrémentale
- Centrée sur l'architecture
- Orientée par la réduction des risques



Les principes clés de l'A00 (1/2)

- **Guidé par les cas d'utilisation :**
 - Le projet est mené en tenant compte des besoins et des exigences des utilisateurs.
 - Les cas d'utilisation du système sont identifiés, décrits avec précision et priorisés.
- **Piloté de façon itérative et incrémentale :**
 - Le projet est découpé en itérations de courte durée (environ 2 semaines) qui permettent de mieux suivre l'avancement global.
 - A la fin de chaque itération, une partie exécutable (release) du système final est produite, de façon incrémentale.

Les principes clés de l'A00 (2/2)

- **Centré sur l'architecture :**
 - Tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées.
 - Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML ou graphiquement, et pas seulement documentée en texte.
- **Orienté par la réduction des risques :**
 - Les risques majeurs du projet doivent être identifiés au plus tôt mais surtout levés le plus rapidement possible.
 - Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.

Plan

- Analyse Orientée Objet (AOO)
- **Étape 1: Cas d'utilisation**
- Étape 2: Diagramme de séquence “système”
- Étape 3: Diagramme de classes d'analyse

1. Identifier le contexte

Il s'agit de:

- Considérer le futur logiciel comme une boîte noire qui doit fournir des services à son environnement. Par environnement, on entend les utilisateurs qui ont besoin de ce logiciel. L'environnement constitue le contexte.
- Se poser la question : à qui le logiciel devra servir ?
- Recueillir les besoins de l'utilisateur
 - noyés dans une grande quantité d'informations,
 - exprimés de façon non structurée,
 - sans forte cohérence (oublis, contradictions, imprécisions, etc).
- Dans UML, on appelle ce qu'on doit analyser, concevoir et réaliser : le **système**.



Description textuelle du logiciel, des besoins et des exigences

2. Identifier les acteurs

Il s'agit de répondre aux questions suivantes:

*nom
commun*

- Qui utilise, installe, démarre, arrête, maintient le logiciel?
- Qui donne et/ou reçoit des informations du logiciel?
- Quels autres systèmes utilisent les services proposés par le logiciel?



Une liste préliminaire des acteurs du système

3. Identifier les cas d'utilisation

Il s'agit de répondre aux questions suivantes:

Verbe à
l'infinitif

- Qu'est ce que l'acteur attend du logiciel?
- Est-ce que le logiciel enregistre des informations? Quels acteurs les créent, les consultent, les modifient ou les détruisent?
- Est-ce qu'un acteur indique au système des changements dans son état?
- Y-a-t-il des événements externes que le logiciel doit connaître? Quels acteurs l'en informent?



Une liste préliminaire des cas d'utilisation par acteur du système

4. Réaliser les diagrammes de cas d'utilisation

- Avec les diagrammes de cas d'utilisation, on se place clairement du côté des utilisateurs.
- Il faut rester au niveau des grandes fonctions du système.
- L'ensemble des cas d'utilisation doit couvrir exhaustivement tous les besoins fonctionnels du système.
- Il ne doit pas y avoir de notion temporelle dans un diagramme de cas d'utilisation. Il ne faut pas se dire que l'acteur fait ceci, puis le système lui répond cela, ce qui implique une réaction de l'acteur, et ainsi de suite.



Un diagramme de cas d'utilisation en transcrivant les réponses précédentes sur un schéma qui montre les cas d'utilisation (ovales) reliés par des associations (lignes) à leurs acteurs principaux (icône du « stick man »).

5. Décrire textuellement les cas d'utilisation

- Bien que de nombreux diagrammes d'UML permettent de décrire un cas d'utilisation, il est recommandé de rédiger dans certains cas, une description textuelle car c'est une forme souple qui convient dans bien des situations.
- La description textuelle permet de prendre en compte le séquençement temporel des cas.
- L'intérêt des cas d'utilisation ne se limite pas au recueil des besoins. La description des cas d'utilisation peut servir de base de travail pour établir les tests de vérification du bon fonctionnement du système, et orienter les travaux de rédaction de la documentation à l'usage des utilisateurs.

 *Une description textuelle des cas d'utilisation*

6. Organiser les cas d'utilisation

- En les regroupant en packages par acteur ou par domaine fonctionnel.
- En ajoutant des relations d'inclusion, d'extension et de généralisation entre cas d'utilisation ;



Une Organisation et une structuration des cas d'utilisation



Plan

- Analyse Orientée Objet (AOO)
- Étape 1: Cas d'utilisation
- **Étape 2: Diagramme de séquence “système”**
- Étape 3: Diagramme de classes d'analyse



Diagramme de séquence système (1/2)

Il s'agit d'un diagramme de séquence avec l'acteur principal à gauche, puis un objet unique représentant le système en boîte noire (la représentation d'un système sans considérer son fonctionnement interne), et, enfin, les éventuels acteurs secondaires sollicités durant le scénario à droite du système.

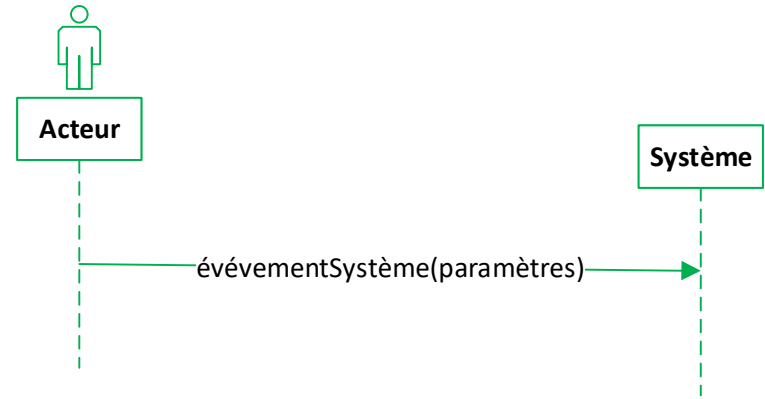
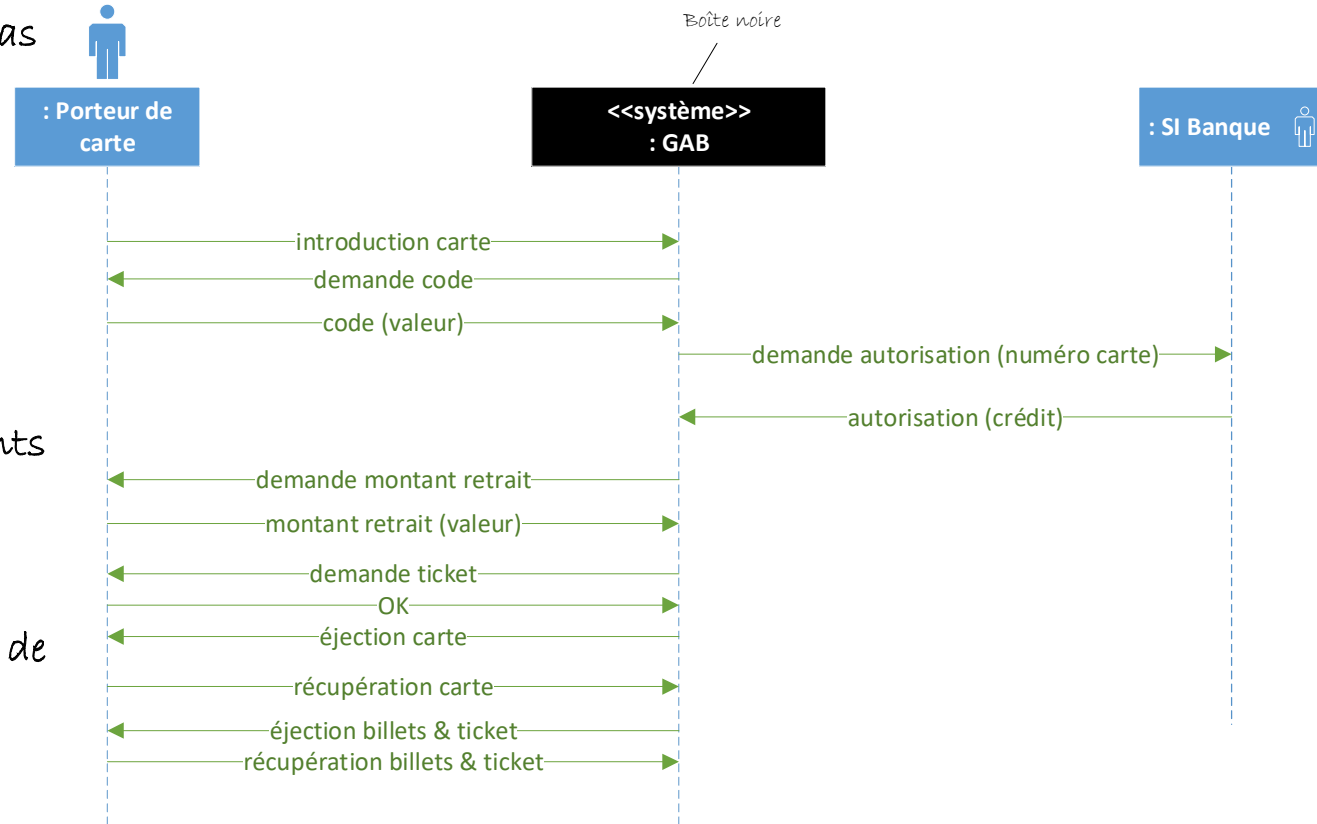


Diagramme de séquence système (2/2)

- Le diagramme de séquence système (DSS) constitue **une excellente solution pour les descriptions dynamiques d'un cas d'utilisation.**
- Le diagramme de séquence «système » illustre la succession temporelle des événements causés par des messages venant des acteurs.

Diagramme de séquence système – Retirer de l'argent

- A partir du diagramme de cas d'utilisation on crée le diagramme de séquence système.
- Le système est considéré comme une boîte noire.
- Il faut nommer les événements système avec des termes de niveau utilisateur, non en termes technologiques. Par exemple code(valeur) au lieu de taperLeCodeSurleClavier.



Pour information

- Il faut faire la différence entre le Diagramme de Séquence d'Analyse (DSA) et le Diagramme de Séquence Système (DSS).
- Le Diagramme de Séquence d'Analyse (DSA) détaille le Diagramme de Séquence Système (DSS).
- Aussi bien le DSA que le DSS aident à l'identification des principales classes d'analyse.
- Dans le cadre de cours nous considérerons uniquement le DSS.



Plan

- Analyse Orientée Objet (AOO)
- Étape 1: Cas d'utilisation
- Étape 2: Diagramme de séquence "système"
- **Étape 3: Diagramme de classes d'analyse**



À retenir

Dans ce cours nous présenterons de nombreuses techniques celle que vous devriez maîtriser est incontestablement:

- **“Comment attribuer les bonnes responsabilités aux bons objets en utilisant des principes réutilisables” !**
- **Pourquoi ?**
 - Quelles que soient les étapes d'analyse ou de conception que vous dérouliez, l'attribution des responsabilités est une étape inévitable.
 - L'attribution des responsabilités a un impact majeur sur la robustesse, la maintenabilité et l'évolutivité du système.
 - Cette étape détermine le résultat de votre analyse et de votre conception.

Diagramme de cas d'utilisation vs Diagramme de classes

DIAGRAMME DE CAS D'UTILISATION

- Langage de l'utilisateur
- Vue externe du système
- Cas d'utilisation
- Contrat utilisateur/développeur
- Peut être redondant/inconsistant
- Capture les fonctionnalités
- Spécifie des cas d'utilisation

DIAGRAMME DE CLASSES

- Langage du développeur
- Vue interne du système
- Classes
- Vient des développeurs
- Ni redondant / ni inconsistant
- Réalisation des fonctionnalités
- Spécifie des realisations de cas d'utilisation

Diagramme de classes d'Analyse

Le diagramme de classes est le point central dans un développement orienté objet.

En analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs.

Qu'est ce qu'une classe d'analyse ?

- **Une classe d'analyse est soit un concept du métier des utilisateurs...**
 - Son existence est indépendante de l'informatisation de l'activité.
 - Elle peut être identifiée à partir d'une analyse générale du domaine.
 - **Client, Compte, Virement ...**
- **...soit un concept induit par l'informatisation**
 - Elle décrit souvent une interface externe ou représente un acteur.
 - page relevé de compte...
 - Elle représente la dynamique d'un cas d'utilisation du système.
 - contrôleur de comptes...
 - On parle aussi de classe applicative

Classe pertinentes en analyse OO?

Parmi les classes candidates il faut écarter celles...

- qui sont **redondantes** – synonymie
- qui sont **trop générales**
- qui sont **trop spécifiques**
- qui représentent une **valeur** – attribut
- qui représentent un **comportement** – méthode



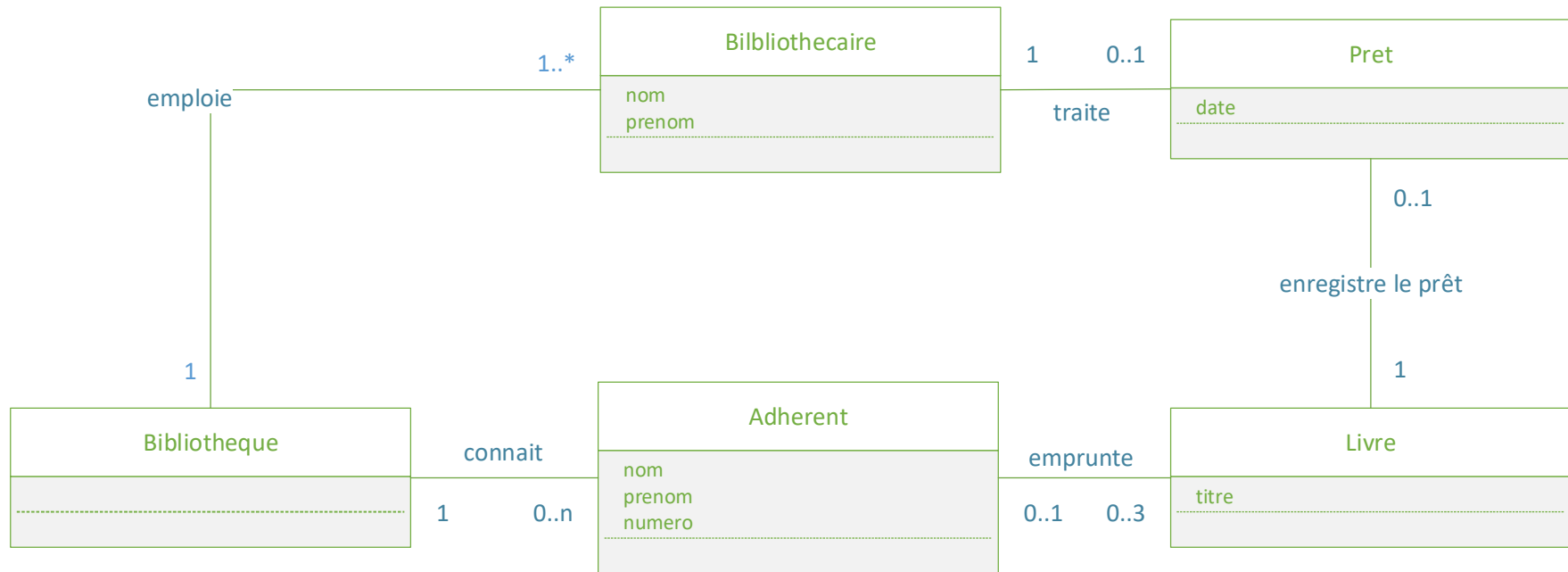
Classe pertinentes en analyse OO? (suite)

... et s'assurer de l'intérêt de celles

- qui représentent un **rôle** d'une entité - rôle sur une relation
- qui représentent un **acteur** – substitut
- qui représentent des **groupes d'objets** - conteneurs



Diagramme de classes d'analyse du S.I. d'une bibliothèque



A partir du diagramme de cas d'utilisation on crée le diagramme de classes d'analyse

Les packages d'analyse

- Le package d'analyse
 - est un regroupement de classes et d'associations;
 - qui permet de répartir le travail d'analyse;
 - qui doit rester stable au travers des différentes itérations.
- Il peut contenir d'autres packages.
- Il peut dépendre des informations détenues par d'autres packages.
- Les packages constitue un composant de l'analyse orientée objet
 - chaque classe ou association n'appartient qu'à un package;
 - le nom complet d'un élément est : nomPackage::nomElément.

Si l'analyse du système est sautée ou négligée...

- Le manque de connaissances « métier » peut provoquer des erreurs ou des délais importants.
- Vous allez résoudre le mauvais problème!
- Vous allez mélanger le « quoi » et le « comment ».
- Vous ne pourrez pas facilement changer de technologie.



Espace du problème

Espace de la solution

Analyse et Conception Orientée Objet

Conception Orientée Objet