

Support de cours en système d'informations, méthodes avancées : Chapitre 2 : Etude préliminaire

1. Introduction

L'étude préliminaire permet de recueillir des informations initiales sur le système d'information à développer. Il s'agit dans cette phase de définir le contour du système, les différents acteurs, ainsi que les messages d'interaction entre les acteurs et le système.

2. Elaboration du cahier des charges (CdC)

Le CdC est un document qui rassemble les spécifications du projet par le maître d'ouvrage (entreprise cliente) pour le maître d'œuvre (ex :boite informatique, entreprise de développement). A cette étape s'effectue l'identification de l'entreprise, ses activités, et les activités concernées par le projet. Le CdC doit contenir les choix techniques de l'entreprise. Par exemple

- ☐ La méthode ou approche de développement. **Exemple** : UP/UML,
- ☐ L'architecture de développement / déploiement. **Exemples** : Client/serveur, trois tiers,...
- ☐ L'infrastructure de télécommunication. **Exemples** : réseau privé, Internet...
- ☐ Le langage ou environnement de développement. **Exemples** : Java, ...
- ☐ Le type du SGBD. **Exemples** : SGBDR, base XML...

A partir de CdC, le maître d'œuvre identifie deux types de besoins : **besoins fonctionnels et besoins opérationnels**.

- ☐ **Besoins fonctionnels** : ils concernent le métier, i.e. les tâches relevant du domaine étudié.
Exemple : inscrire les étudiants, établir les documents de scolarité, rechercher un étudiant par son nom ou numéro d'inscription...
- ☐ **Besoins opérationnels** : cela concerne les aspects liés à l'exploitation du système mais indépendants des fonctions. Ils sont d'ordre technique.
Exemple : sécuriser l'application, prendre en charge un grand volume de données à échanger, assurer la disponibilité H24, 7/7 du système...

Remarque : les besoins métier et opérationnels sont exprimés par les utilisateurs du système. Cette expression ne doit pas sous-entendre nécessairement les solutions (informatiques) pour répondre aux besoins.

Exemple pratique : Gestion d'une bibliothèque : le SI concerne la bibliothèque d'une université.

- **Cahier des charges** : le système à développer doit répondre aux besoins fonctionnels suivants
 - Centraliser les informations des livres, prêts et consultations en une seule source et éliminer les incohérences.
 - Permettre l'exécution des tâches de *gestion des prêts* et de gestion des *consultations* internes (ajout de prêt, ajout de restitution, ajout de consultation interne, ajout d'adhérent, ajout

d'adhésion, renouvellement d'adhésion, etc.)

Les besoins opérationnels sont exprimés comme suit :

- *Sécurisation* de l'accès des adhérents et agents à l'application.
- Possibilité, pour les adhérents *d'accéder* au système en *interne* (hall de la bibliothèque) et de *l'extérieur* (Internet) pour la consultation de la liste de livres.

Les grands choix techniques

- Processus de développement : processus de développement suivi (processus 2TUP)
- Langage de modélisation : UML
- Langage de programmation : PHP - SGBD : Oracle
- Architecture : Client/serveur

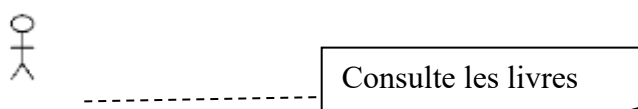
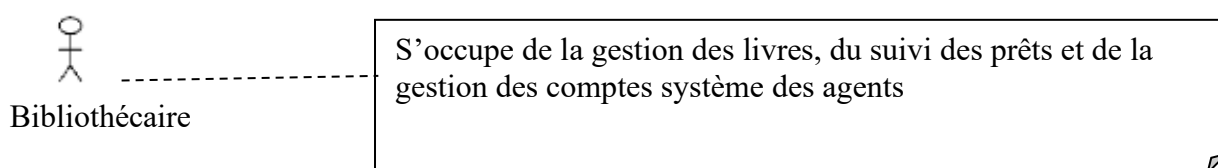
3. Identification des acteurs

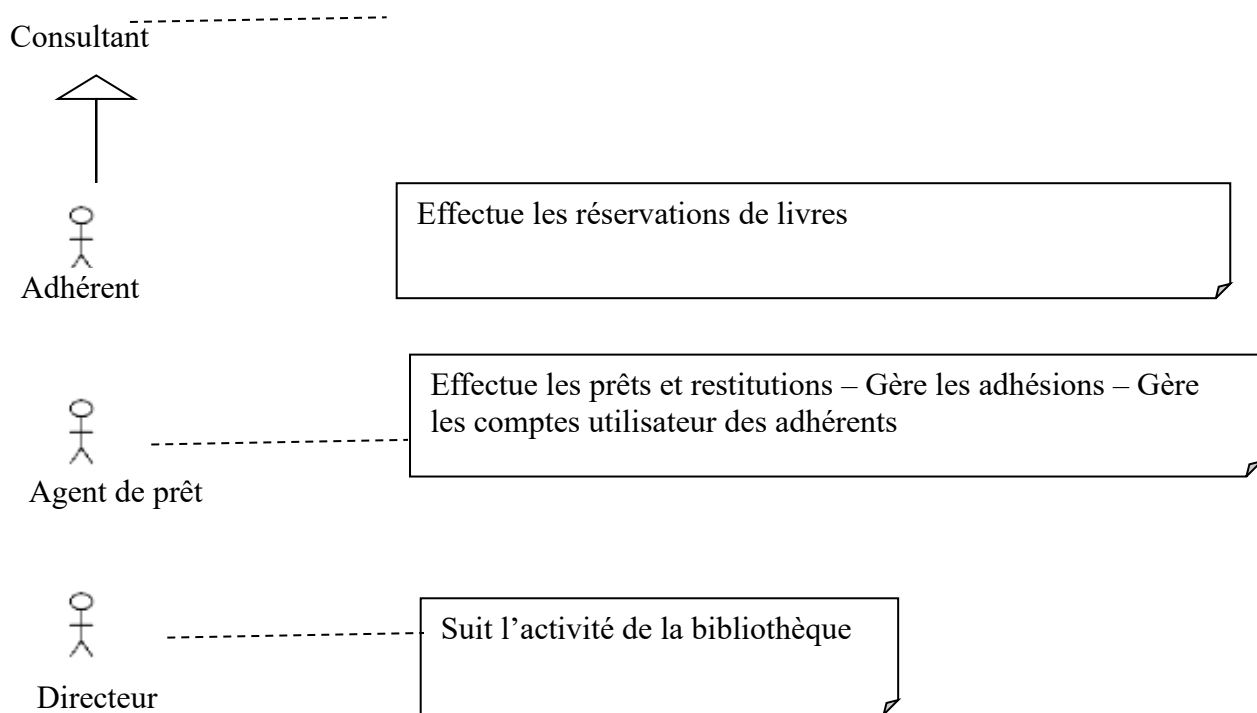
Un acteur est un utilisateur type du système qui représente une **responsabilité** par rapport au système ou un rôle plutôt qu'une personne physique. Un acteur peut être concrétisé par une personne, un système ou une machine.

Un acteur est représenté par le symbole  auquel on peut associer son rôle par rapport au système sous la forme d'une note UML. On peut aussi définir une hiérarchie des acteurs par un mécanisme de spécialisation / généralisation.

Erreurs à éviter : (1) impliquer des acteurs indirects qui n'ont pas d'interaction directe avec le système. (2) identifier des composants du système en tant qu'acteurs.

Exemple pratique : Gestion de la bibliothèque - Identification des acteurs :





4. Identifier les messages

Un message est la spécification d'une communication d'informations entre un émetteur et un récepteur. La notion de message n'est pas limitée à l'échange de messages entre les objets. A ce niveau, elle représente l'échange d'informations entre le système et les différents acteurs. L'acteur envoie des messages d'alimentation du système et en reçoit éventuellement les mêmes messages (en guise de consultation) ou carrément de nouveaux messages (issus de traitements).

Remarque : les messages doivent être identifiés entre chaque acteur et le système, i.e. *il ne faut pas* identifier les messages entre les acteurs eux-mêmes.

Exemple : le système de gestion de la bibliothèque envoie et reçoit plusieurs messages.

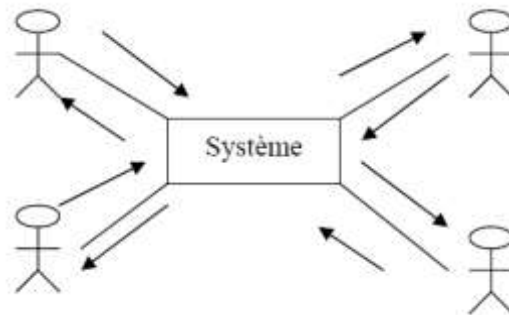
Réception : informations sur les livres, prêts, adhérents, agents, etc.

Envoi : informations sur les dates de restitutions, les adhérents à sanctionner, les dates de fin d'adhésion (pour renouvellement), etc.

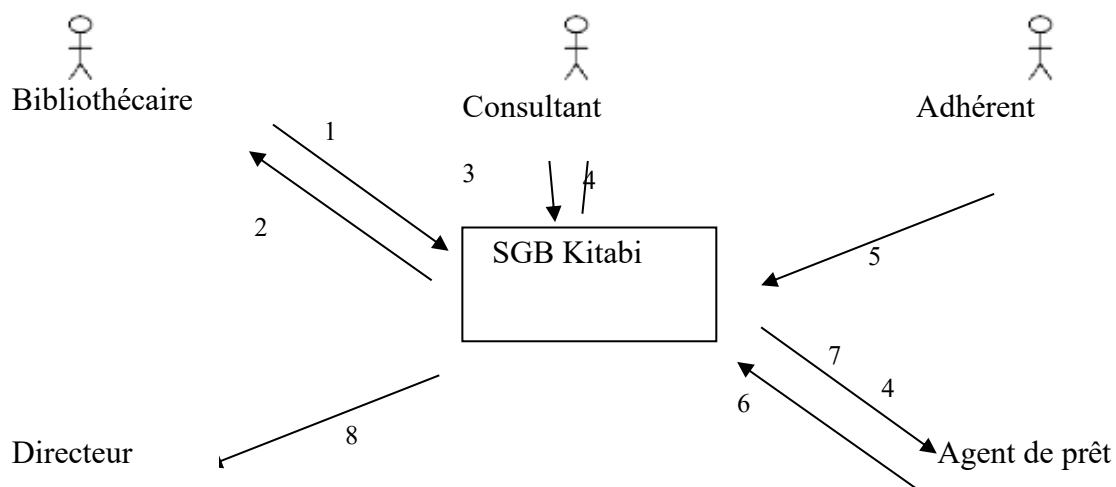
5. Modéliser le contexte

Tous les messages (système <--> acteurs) identifiés dans l'étape précédente peuvent être présentés de façon synthétique sur un diagramme, que l'on peut qualifier de diagramme de contexte dynamique. On utilise un diagramme de collaboration de la façon suivante :

- Le système étudié est représenté par un objet central ;
- Cet objet central est entouré par un autre objet symbolisant les différents acteurs.
- Des liens relient le système à chacun des acteurs ;
- Sur chaque lien sont montrés les messages en entrée et en sortie du système.



Exemple : diagramme de contexte dynamique de système de gestion de la bibliothèque



Numéro du message	Nature des messages
1	Information sur les livres Informations sur les agents de prêt
2	Livres trop demandés Livres aux exemplaires insuffisants
3	Livres recherchés
4	Informations sur les livres et leurs disponibilités
5	Informations de reservation (livre, date de

	réservation) – annulation de réservation
6	Informations sur les adhérents – informations sur les prêts et restitutions – informations sur les comptes d'utilisateurs
7	Adhérents sanctionnés, informations détaillées sur les prêts
8	Taux de renouvellement, taux de prêt de livres