

Informatique Repartie

Présentation Projet 2020/2021

Cecilia Zanni-Merk

cecilia.zanni-merk@insa-rouen.fr

Bureau BO B R1 04

Agenda

- Rappels UML
- Démarche de mise en œuvre pour le projet
- Livrables
- Présentation du projet
- Organisation et deadlines

Rappels UML

Très rapidement

Rappels UML

- UML permet d'exprimer **la structure, la fonction et le comportement** d'un programme

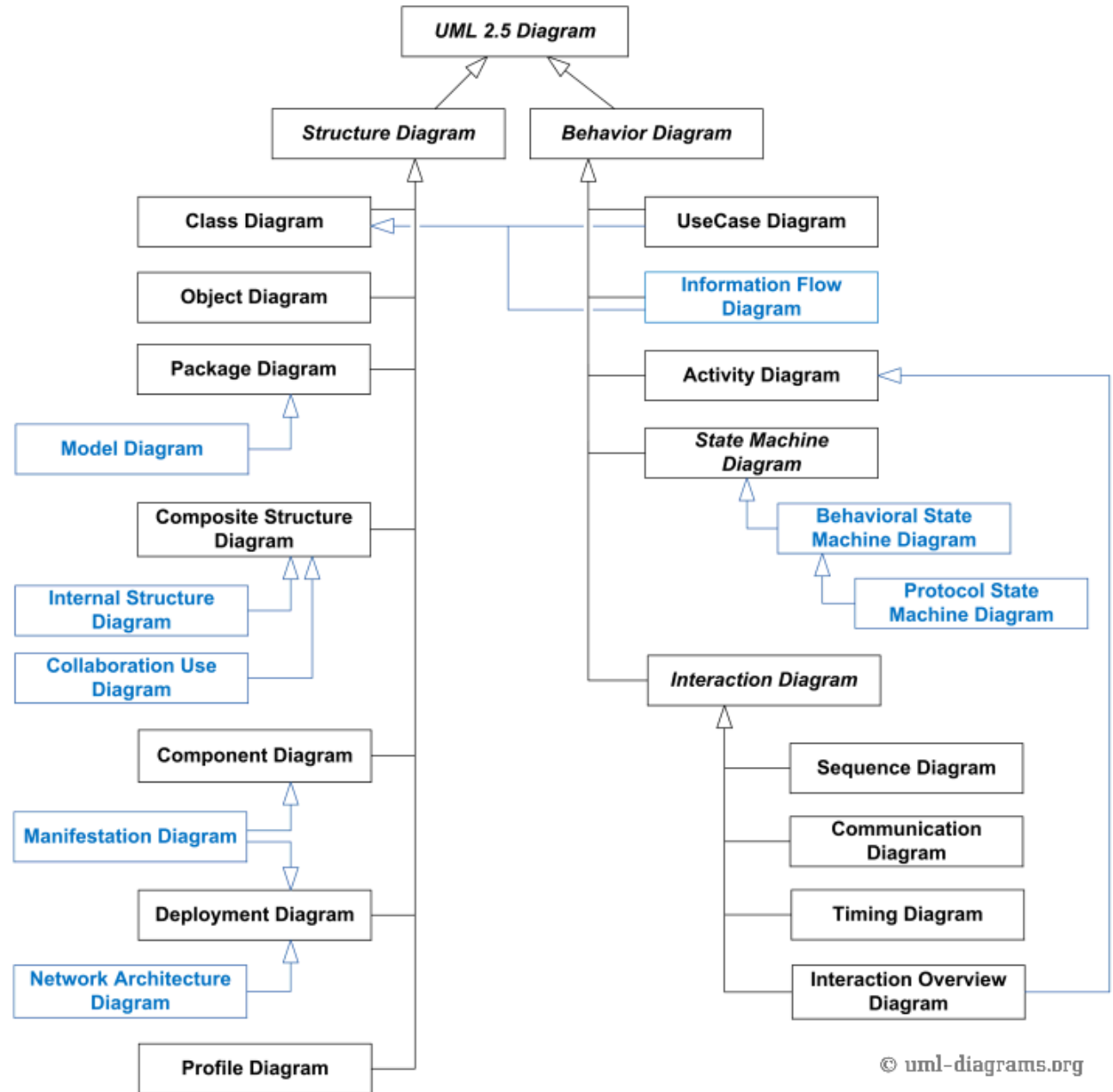


Structure (données) QUOI ?

Fonction COMMENT?

Comportement QUAND?

Rappels UML 2.5



UML 2.5 propose 14 diagrammes standard

- Sept diagrammes de structure
 - De classes
 - De structure composite
 - D'objets
 - De composants
 - De paquetages
 - De déploiement
 - De profils
- Sept diagrammes de comportement
 - De cas d'utilisation
 - D'interaction
 - D'activités
 - De communication
 - De séquence
 - De temps

UML 2.5 propose 14 diagrammes standard

- Sept diagrammes de structure
 - De classes
 - De structure composite
 - D'objets
 - De composants
 - De paquetages
 - De déploiement
 - De profils
- Sept diagrammes de comportement
 - De cas d'utilisation ← FONCTION
 - D'interaction
 - D'activités
 - De communication
 - De séquence
 - De temps

UML 2.5 propose 14 diagrammes standard

- Sept diagrammes de structure

- De classes ← spéc / conception
- De structure composite
- D'objets
- De composants ← conception
- De paquetages
- De déploiement ← conception
- De profils

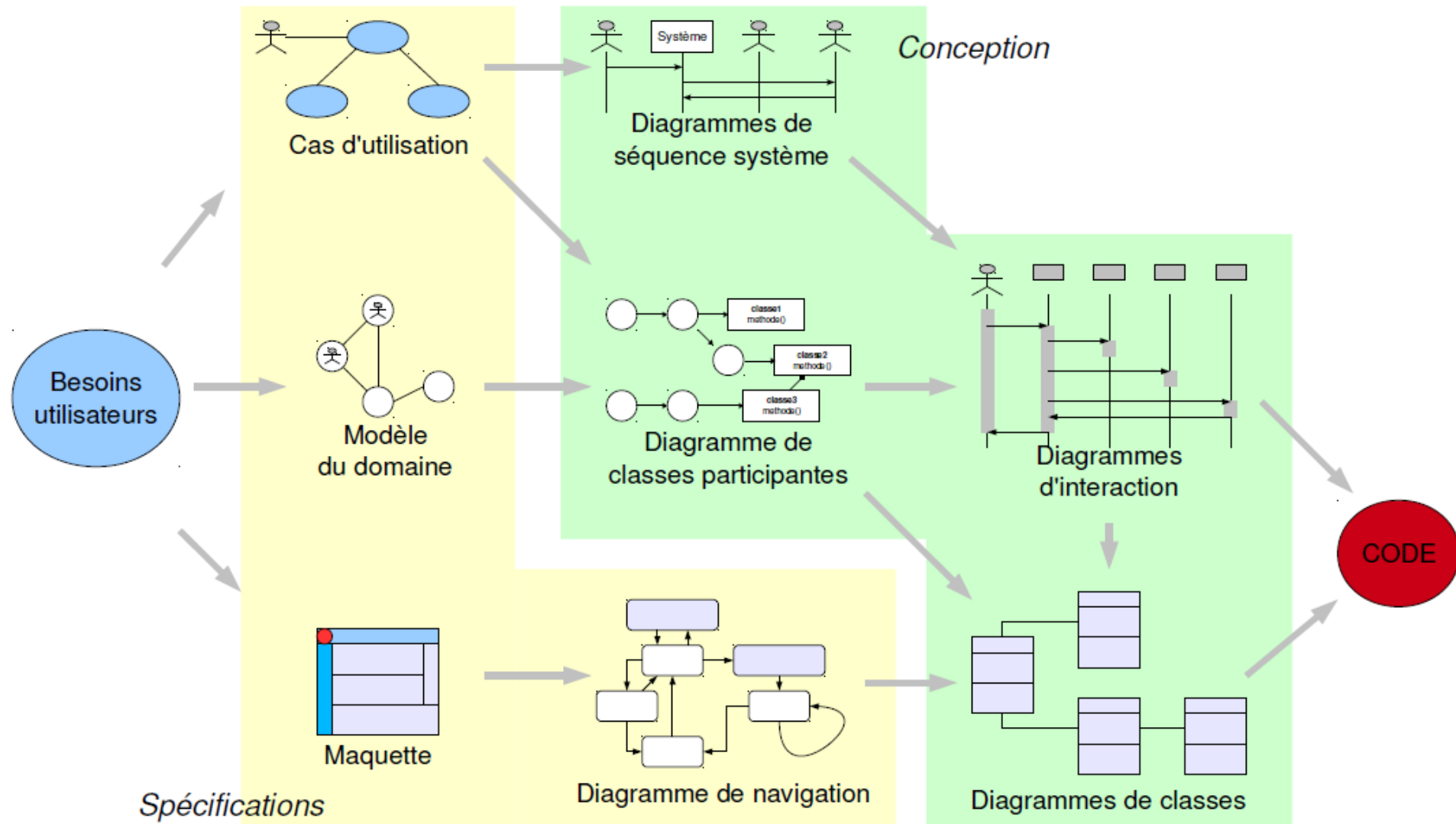
- Sept diagrammes de comportement

- De cas d'utilisation ← spéc
- D'interaction
- D'activités
- De communication
- De séquence système ← spéc
- De temps

Démarche de mise en oeuvre

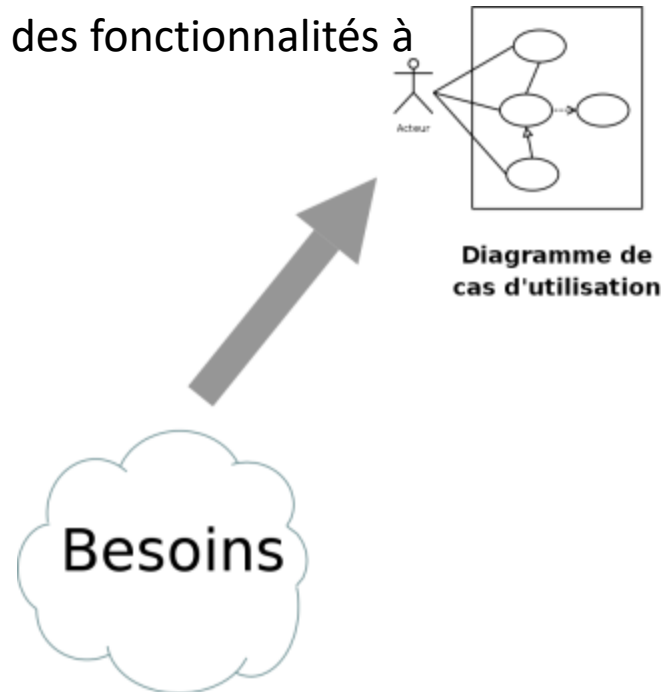
<https://laurent-audibert.developpez.com/Cours-UML/?page=mise-en-oeuvre-uml>

Démarche de GL simplifiée

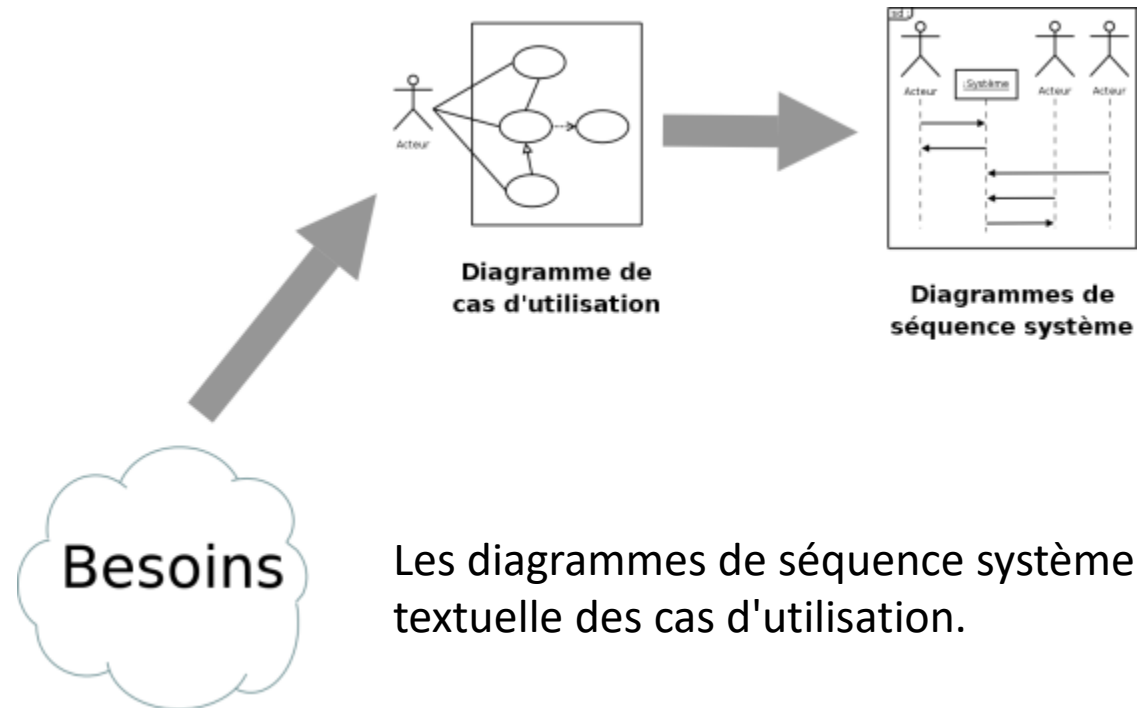


Identification des besoins

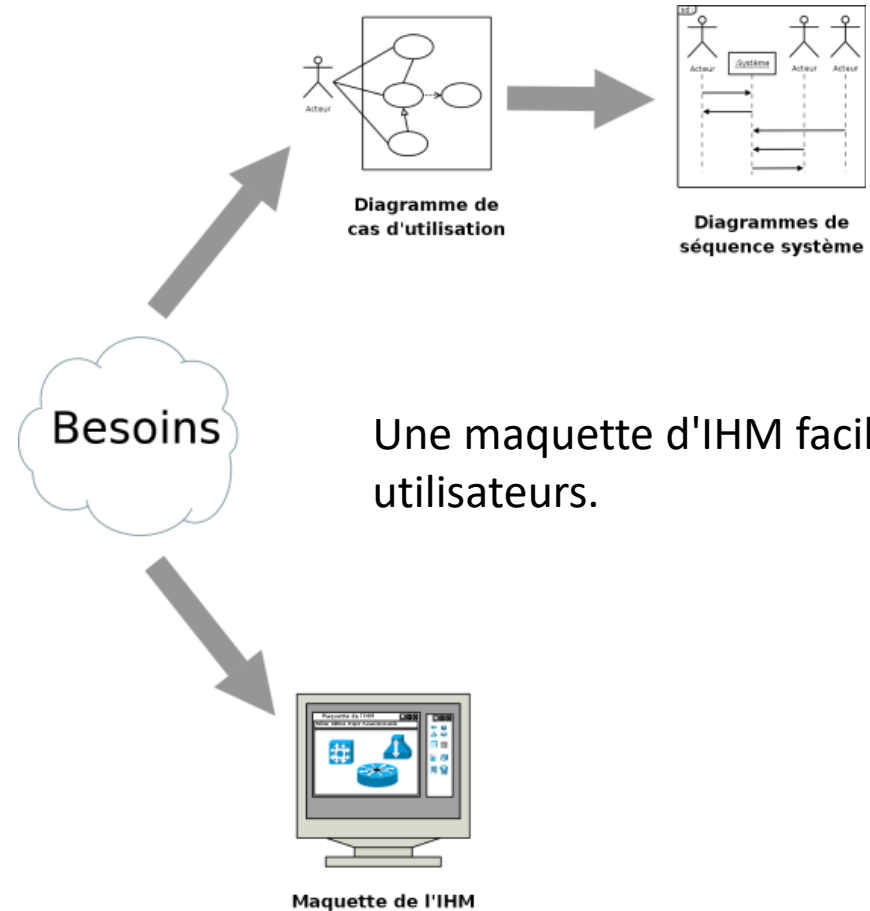
Les besoins sont modélisés par un diagramme de cas d'utilisation, à partir de la liste exhaustive des fonctionnalités à développer



Identification des besoins

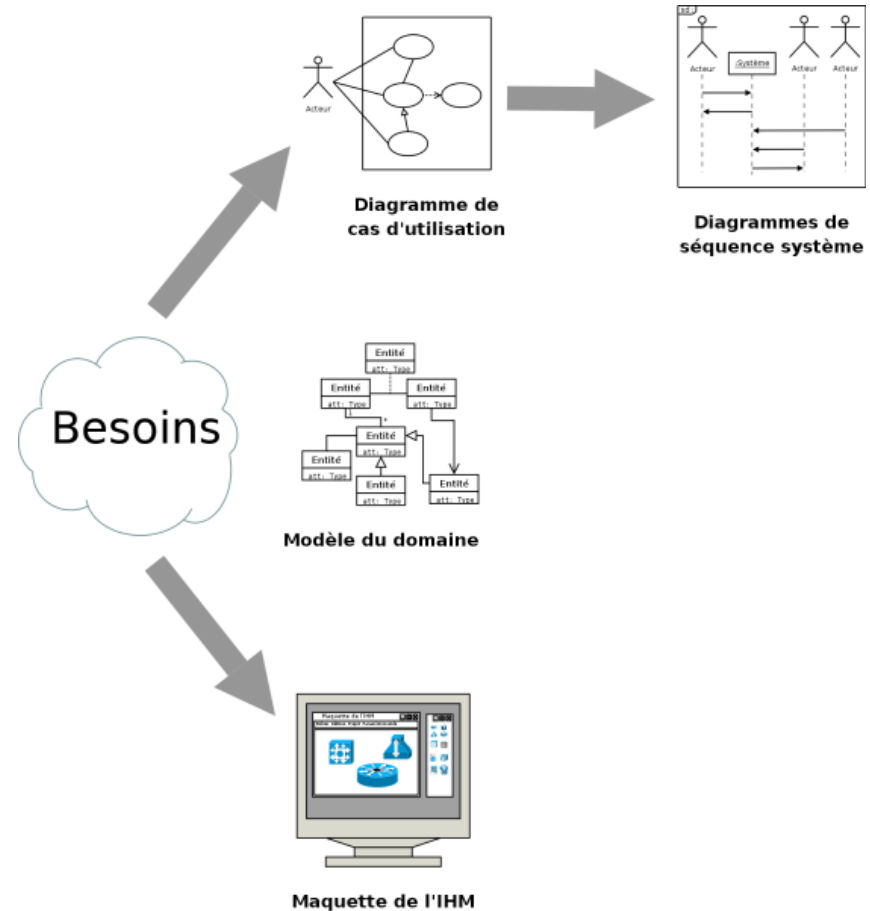


Identification des besoins

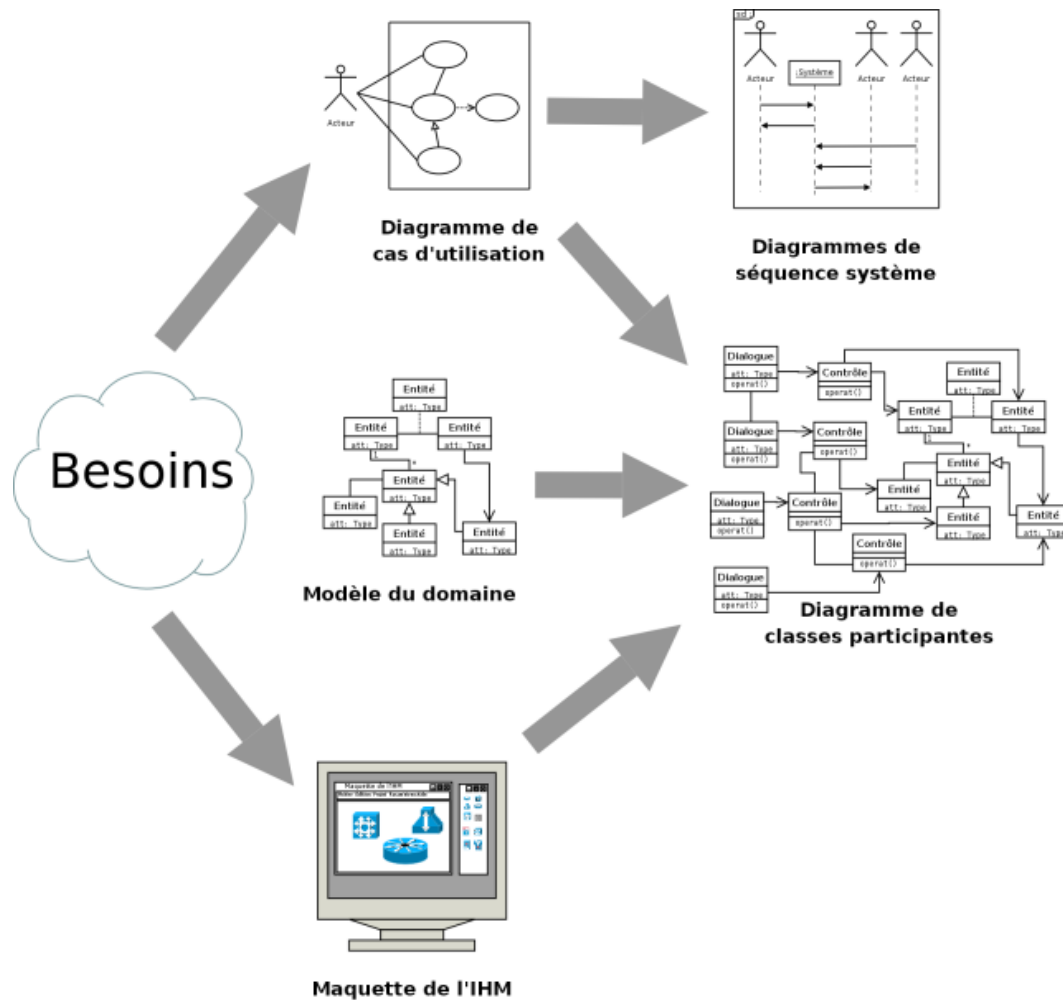


Phase d'analyse

La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes.



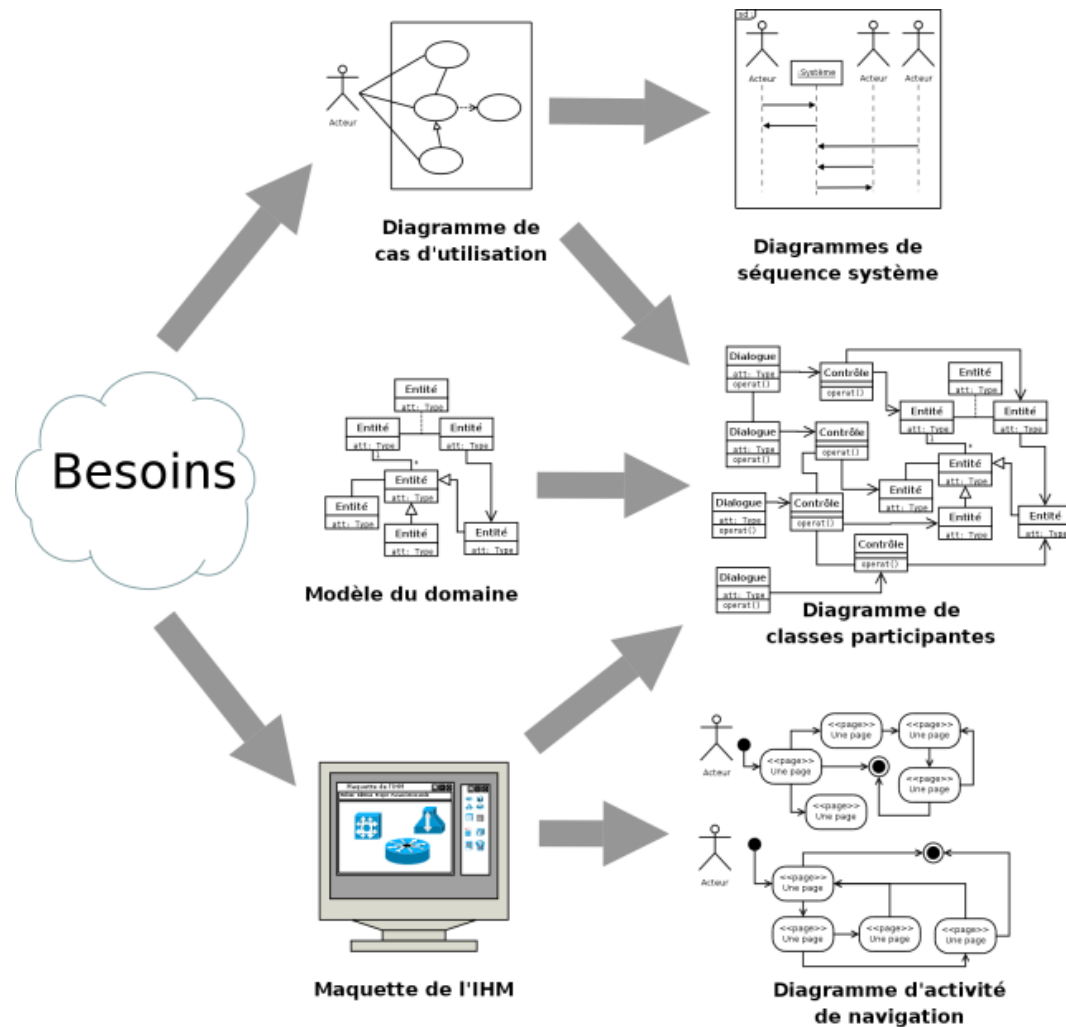
Phase d'analyse



Le diagramme de classes participantes effectue la jonction entre les cas d'utilisation, le modèle du domaine et les diagrammes de conception logicielle.

On inclura ici les éléments issus du modèle MVC ... les dialogues, les contrôles et les entités (qui proviennent directement du modèle du domaine), ainsi que leurs relations

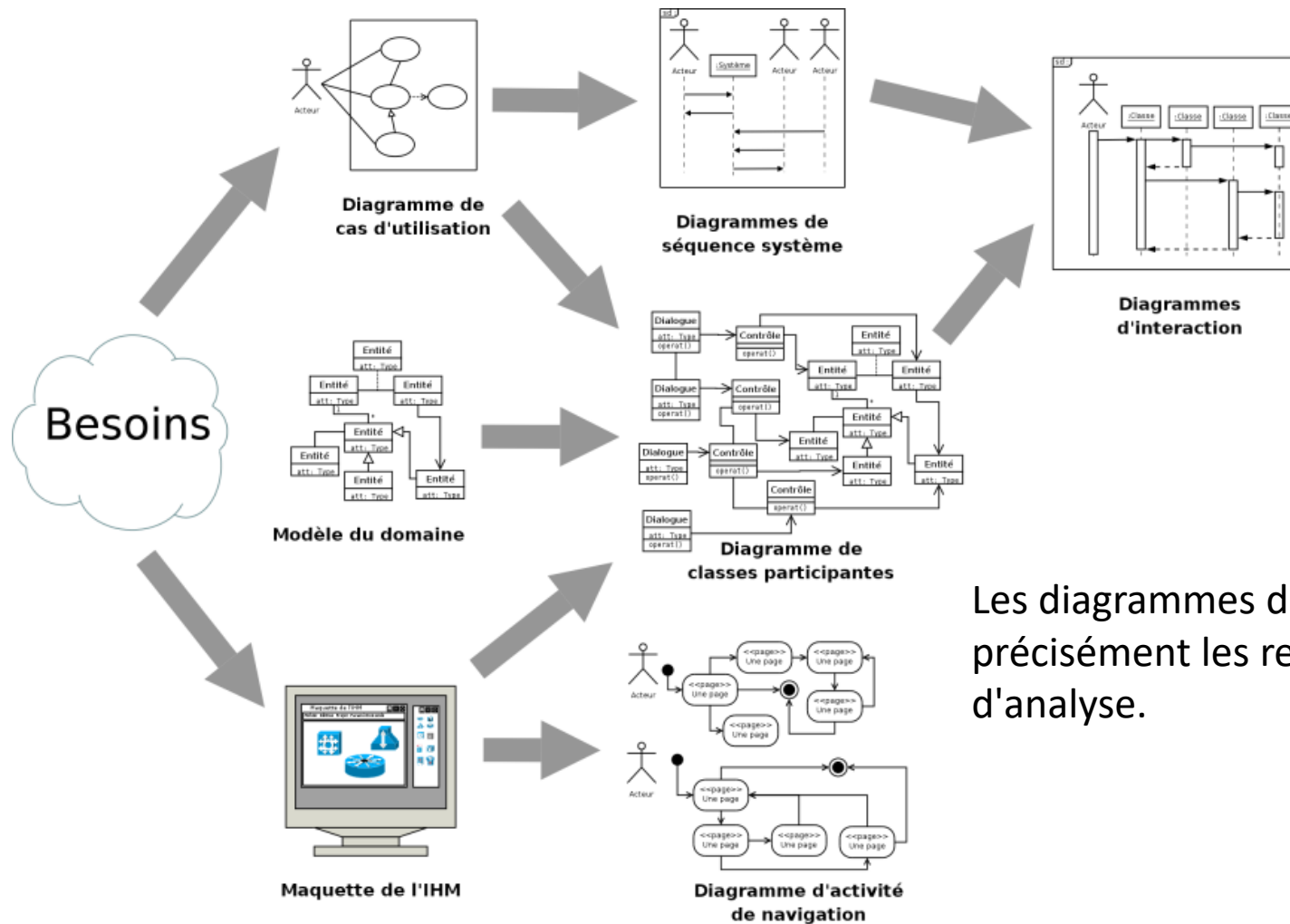
Phase d'analyse



Les diagrammes d'activités de navigation représentent graphiquement l'activité de navigation dans l'IHM.

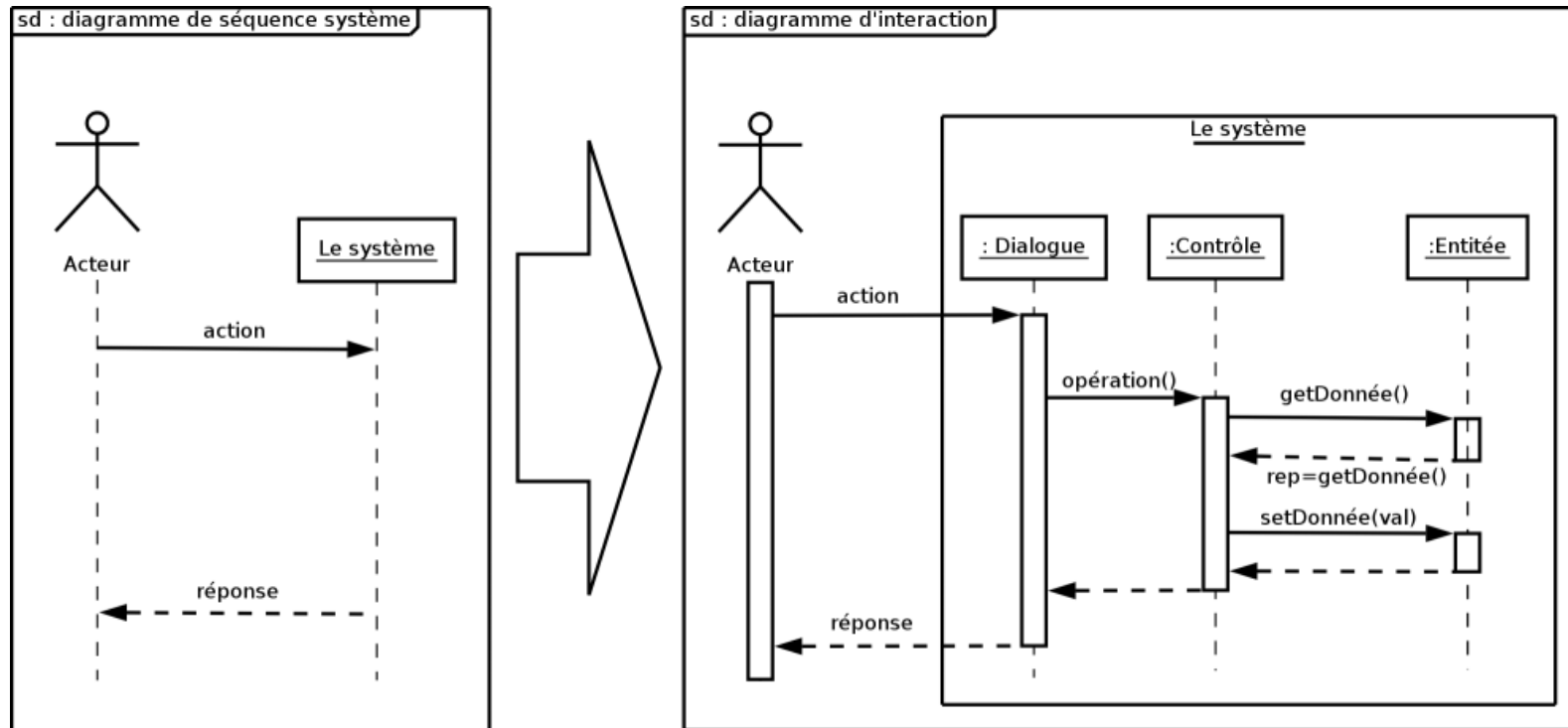
La modélisation de la navigation devra être structurée par acteur

Phase de conception



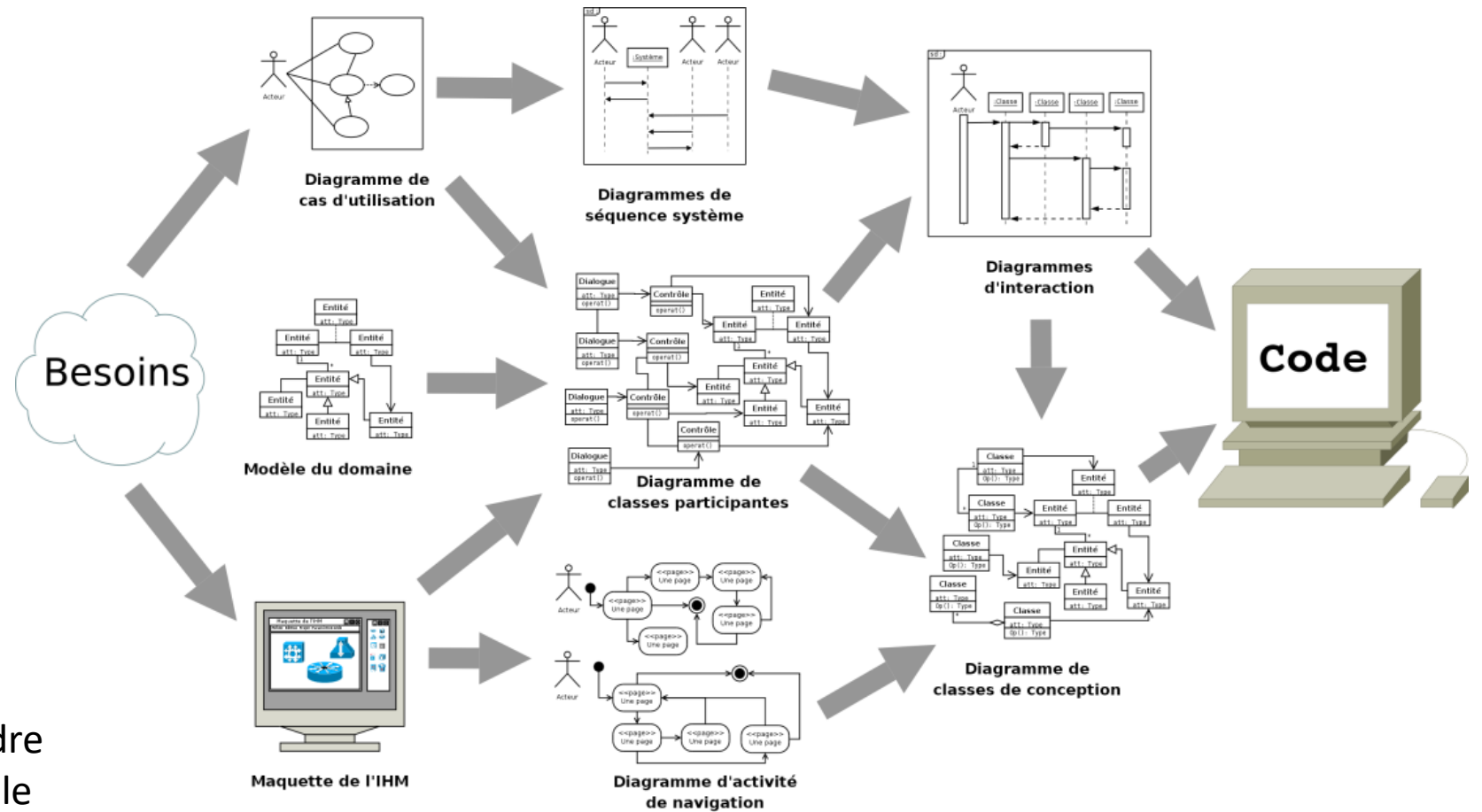
Les diagrammes d'interaction permettent d'attribuer précisément les responsabilités de comportement aux classes d'analyse.

Phase de conception



Le système des diagrammes de séquences système, vu comme une boîte noire, est remplacé par un ensemble d'objets en collaboration.

Phase de conception



Le diagramme de classes de conception servira pour l'implémentation. Il faudra prendre en compte les choix techniques (le langage de programmation, les bibliothèques utilisées ...)

Livrables pour projet

Document de spécifications

- Introduction

- Il s'agit d'une description informelle du projet et de son contexte.
- On doit y trouver notamment comme informations :
 - la liste des fonctions principales,
 - Les différents utilisateurs et leurs caractéristiques,
 - les contraintes matérielles et logicielles.

- Besoins détaillés

- C'est la partie contractuelle à proprement parler puisqu'elle formalise le besoin.
- Elle consiste 3 parties distinctes :
 - les spécifications fonctionnelles (***la liste exhaustive des fonctionnalités à développer, en français 😊***)
 - les spécifications d'interfaces (***avec d'autres logiciels, utiles dans un cadre d'InfoRep***)
 - les spécifications opérationnelles (performance, sécurité, ...)
- Ces différents éléments peuvent s'appuyer en UML sur des **diagrammes de cas d'utilisation**, des **diagrammes de séquence système**, d'un **diagramme de modèle du domaine**, de **maquettes** et d'un **diagramme de navigation**, en fonction des besoins.

Document de conception

- Trois sections: Introduction, Conception préliminaire, Conception détaillée
- Introduction
 - Il s'agit d'une description informelle du projet et de son contexte. Elle est souvent relativement similaire à l'introduction d'un document de spécification.
 - On doit y donc y trouver notamment comme informations :
 - la liste des fonctions principales,
 - les différents utilisateurs et leurs caractéristiques,
 - les contraintes matérielles et logicielles.

Document de conception

- Conception préliminaire
 - Cette étape consiste à réaliser une conception macroscopique, c'est-à-dire permettant de mener à un découpage en packages avec les signatures externes de chaque package. Cette étape peut s'appuyer en UML sur :
 - Un diagramme de modèle du domaine (si non spécifié) ;
 - Des diagrammes de séquence système (si non spécifiés) ;
 - Des maquettes (si non spécifiées) ;
 - Des diagrammes d'activités de navigation (si non spécifiés) ;
 - Des **diagrammes d'interaction** ;
 - Un **diagramme des classes de conception préliminaire** ;
 - Un **découpage de composants avec les interfaces requises et fournies**
 - Un **diagramme de déploiement**

Document de conception

- Conception détaillée

- Cette étape consiste à détailler par package les éléments constituant les composants.
- L'utilisation de **diagrammes de paquetages** est possible
- Concrètement, il s'agit principalement de préciser les attributs et méthodes de classe de toutes les classes participantes et de les regrouper dans un diagramme de classes.
- Les méthodes non triviales devront être commentées et voir leur fonctionnement détaillé par du pseudo-code.
- **Chacun des éléments doit indiquer clairement l'auteur qui prendra en charge son développement**

Rapport final

- Doit contenir
 - Spécifications (Document de spécifications mis à jour si nécessaire)
 - Conception (Document de conception mis à jour si nécessaire),
 - Choix techniques justifiés
 - Améliorations possibles (non prévues dans les spéc)

Présentation du projet

MOMs

- Le terme *message-oriented middleware* (MOM), intergiciel à messages en français, désigne une famille de logiciels qui permettent l'échange de messages entre les applications présentes sur un réseau informatique.

Caractéristiques principales des MOMs

- *Communication asynchrone.* L'application émettrice d'un message et l'application réceptrice du message n'ont pas besoin d'être actives en même temps. Une file d'attente reçoit le message de l'application émettrice et le stocke jusqu'à ce que l'application réceptrice vient lire le message.
- *Persistance des messages.* Les messages présents dans les files d'attente peuvent être sauvegardés sur un support physique pour en assurer la conservation en cas de panne.
- *Fiabilité.* Chaque message envoyé par une application fait l'objet d'un accusé de réception par le MOM. Chaque application qui consomme un message envoie un accusé de réception au MOM. Couplé avec la persistance, ce mécanisme permet de garantir qu'aucun message ne sera perdu dans son transfert entre les applications.

Deux modes de fonctionnement

- *Point à point* : une application produit des messages et une application les consomme. Les messages ne sont lus que par un seul consommateur. Une fois qu'un message est lu, il est retiré de la file d'attente.
- *Publish-subscribe* (par abonnement) : les applications consommatrices des messages s'abonnent à un topic (sujet, catégorie de messages). Les messages envoyés à ce topic restent dans la file d'attente jusqu'à ce que toutes les applications abonnées aient lu le message, ils sont effacés après.

Le projet

- On souhaite simuler un MOM (Message Oriented Middleware) en mode client/serveur.
- Un serveur gèrera les files d'attente des messages et les suscriptions
- Les applications souhaitant envoyer ou recevoir les messages seront simulées par des clients de ce serveur.

Organisation et deadlines

Organisation

- Vous travaillerez en **groupes de 3 étudiants** (*avec un max de 2 groupes de 2 étudiants, en fonction du nombre d'étudiants*)
- Les trinômes sont à renseigner dans ce Google Sheet
<https://docs.google.com/spreadsheets/d/19acke3pmuJg6p6IKbTZ9byKLI1seahPeuLgkrtZvY6M/edit?usp=sharing>

Livrables et deadlines

- Document de spécification : **le lundi 22 mars à 20h00**
- Document de conception : **le lundi 05 avril à 20h00**
- Rapport final et code source : **le lundi 10 mai à 20h00**
- Soutenances : **à organiser**
 - Pendant la semaine d'examen ?
 - Avant ?
 - Après ?