

# UML

## Modélisation générale

# UML : rappel

- UML est un langage capable de décrire n'importe quel système;
- Un diagramme est un modèle;
- Les modèles: diagramme de cas d'utilisation ; diagramme de classes et d'objets ; diagramme d'interaction (diagrammes de séquence, de communication, de timing) ; diagramme d'états-transitions ; diagramme d'activité ; diagramme de composants ; diagramme de déploiement.
- Une vision partielle du système suffit parfois;

# Les démarches

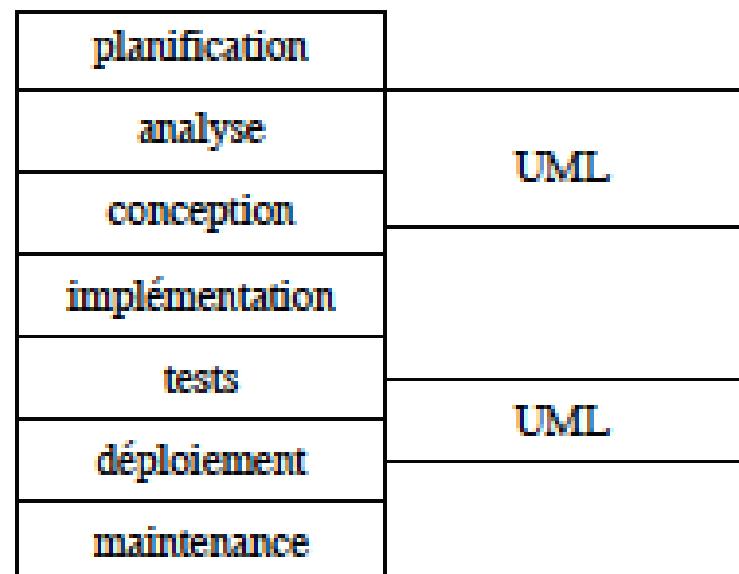
- respecter les étapes du cycle de développement d'un système :
  - planification du projet ;
  - phase d'analyse ;
  - conception ;
  - implémentation ;
  - tests ;
  - déploiement ;
  - maintenance.

# cycle de développement d'un système (1)

- **La planification** - la définition du problème, l'établissement d'un calendrier, la vérification de la faisabilité du projet.
- **L'analyse** - préciser l'étendue des besoins auxquels doit répondre le système, spécifier et de comprendre ce que doit faire ce système (sans se préoccuper de la réalisation).
- **La conception** - comment le système va être réalisé, faire des choix techniques.
- **L'implémentation** - la réalisation du système.
- **Les tests** - vérifier l'adéquation entre la solution et les besoins initiaux avant la mise en production.
- **La maintenance** - conserver en état de marche un système en production.

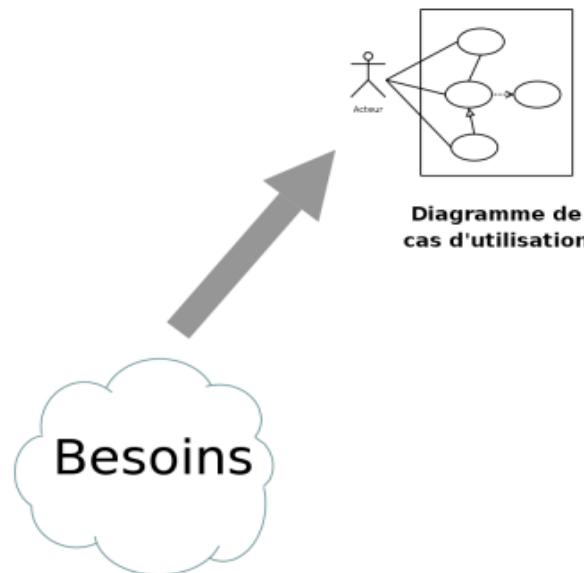
# cycle de développement d'un système (2)

UML intervient dans le plus importants phases.



# Identification des besoins et spécification des fonctionnalités (1)

- Les besoins sont modélisés par un diagramme de cas d'utilisation
- pour identifier et modéliser les besoins des utilisateurs

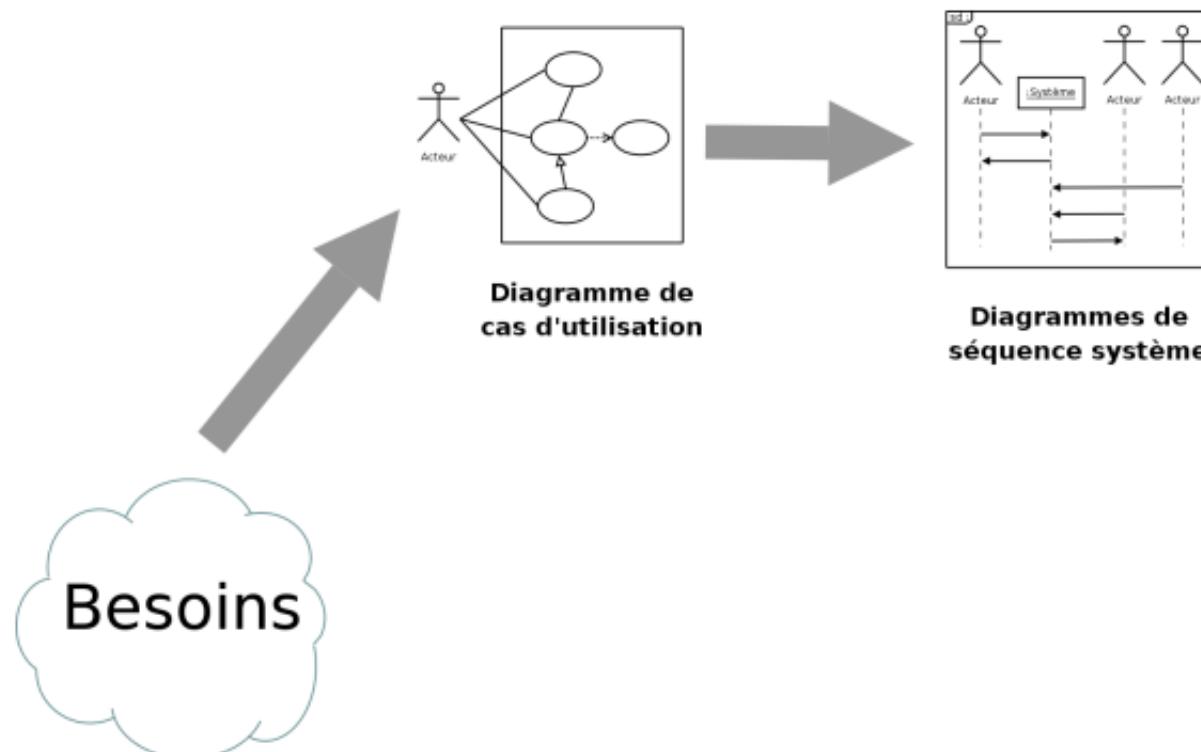


Les interactions entre les acteurs et le système (au sein des cas d'utilisation) seront explicites sous forme textuelle et sous forme graphique au moyen de diagrammes de séquence

# Identification des besoins et spécification des fonctionnalités (2)

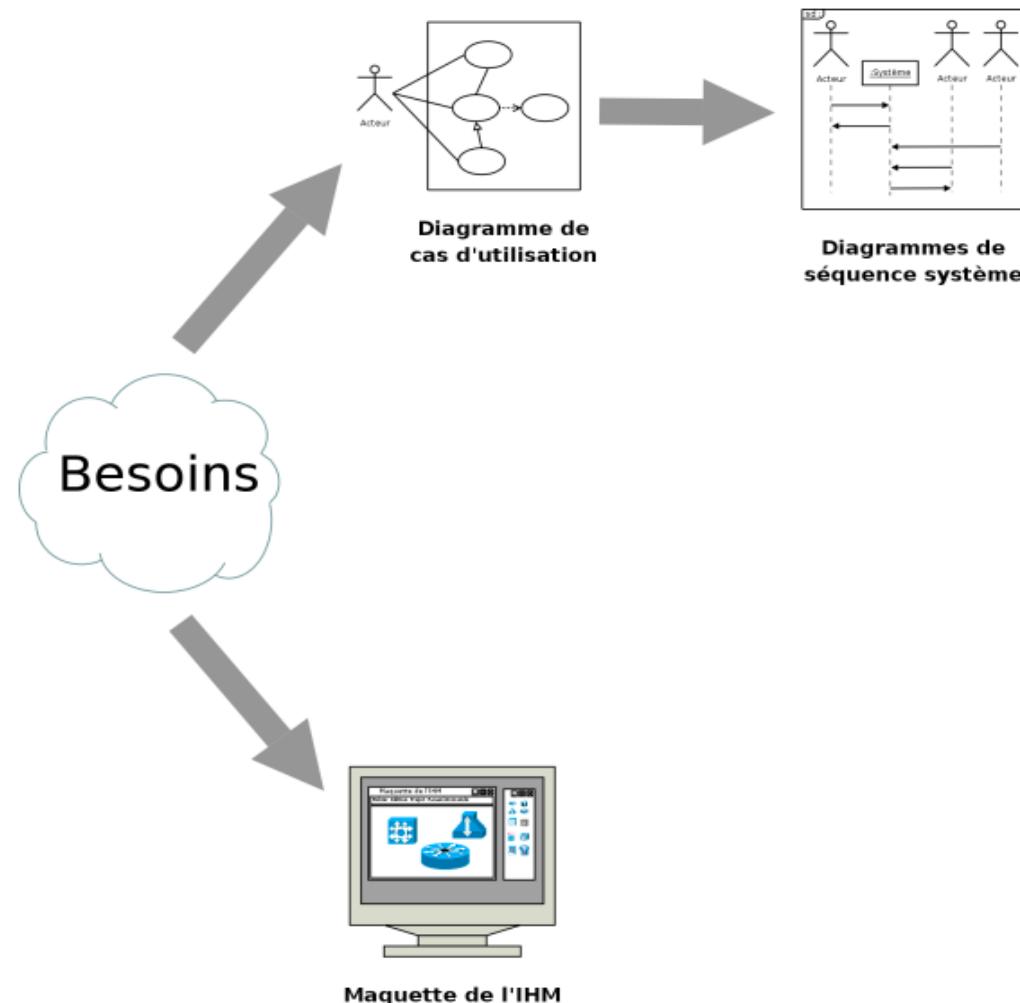
## Spécification détaillée des besoins : diagrammes de séquence système

--> toujours dans la spécification des besoins : détailler la description des besoins par la description textuelle des cas d'utilisation et la production de diagrammes de séquence système



# Identification des besoins et spécification des fonctionnalités (3)

Maquette de l'IHM de l'application



# Phase d'analyse

- La phase d'analyse d'un logiciel se compose de deux parties :
  - l'analyse du domaine de l'application ;
  - l'analyse de l'application.

Les modèles à construire pour la phase d'analyse du domaine de l'application :

Modèle des classes du domaine

Modèle des états du domaine

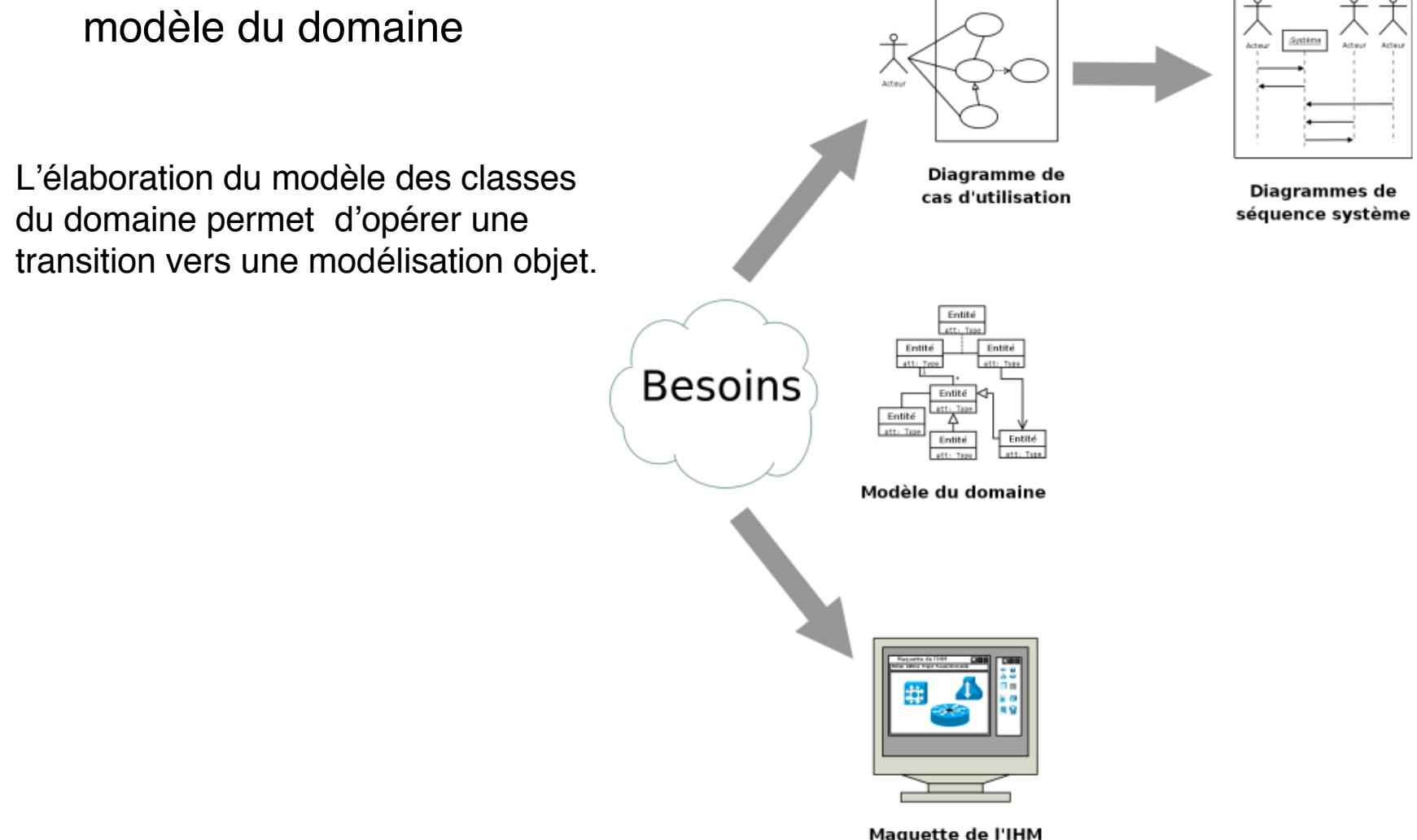
# L'analyse du domaine de l'application (1)

- une première version du modèle de classes est élaborée avec les experts du domaine;
- la construction de diagrammes d'états-transitions pour les classes ayant des états complexes;
- Modèle des classes du domaine :
  - trouver les classes du domaine ;
  - préparer un dictionnaire des données ;
  - trouver les associations entre les classes ;
  - trouver les attributs des classes ;
  - organiser et simplifier le diagramme en utilisant l'héritage ;
  - tester les chemins d'accès aux classes ;
  - itérer et affiner le modèle ;
  - regrouper des classes dans des paquetages.

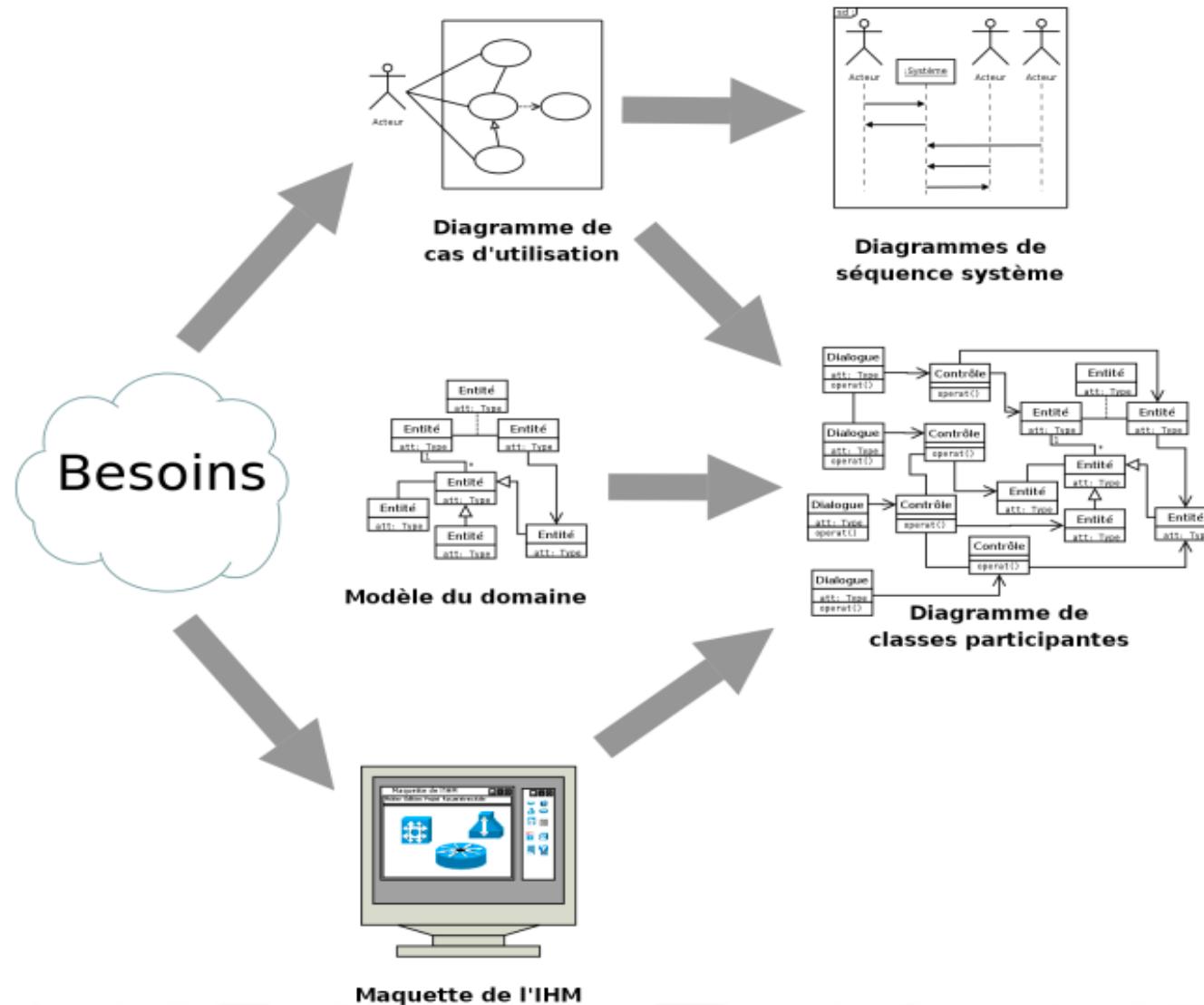
# L'analyse du domaine de l'application (2)

- il est nécessaire de connaître le cycle de vie des objets;
- seules les classes dont les instances ont un cycle de vie complexe doivent donner lieu à des diagrammes d'états-transitions;
- Modèle des états du domaine :
  - identifier des classes du domaine ayant des états complexes ;
  - définir les états ;
  - trouver les événements qui vont engendrer des transitions entre états ;
  - construire les diagrammes d'états.

# L'analyse du domaine de l'application (3)



# L'analyse du domaine de l'application (4)



# Analyse de l'application (1)

- Un diagramme de classes n'offre aucune fonctionnalité aux utilisateurs;
- il faut s'intéresser aux applications qui utilisent ses classes;

Les modèles (diagrammes) pour l'analyse de l'application :

Modèle des interactions de l'application.

Modèle des classes de l'application

Modèle des états de l'application

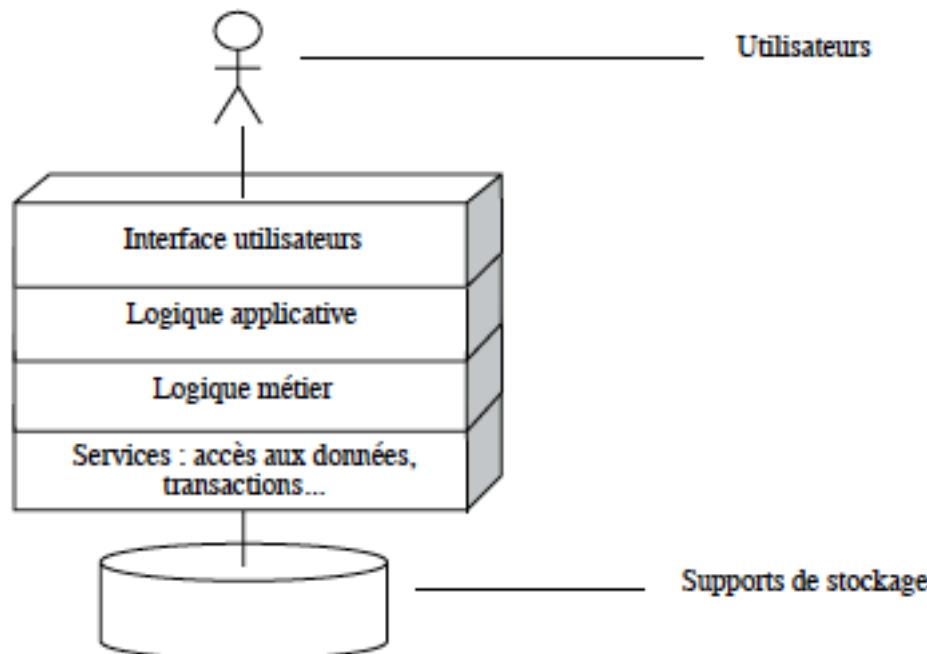
# Analyse de l'application (2)

Modèle des interactions de l'application :

- déterminer les limites du système ;
- trouver les acteurs ;
- trouver les cas d'utilisation ;
- construire le diagramme de cas d'utilisation ;
- préparer les scénarios pour décrire les cas ;
- ajouter des séquences alternatives et des séquences d'exceptions aux scénarios ;
- préparer des diagrammes d'activité pour des cas d'utilisation complexes ;
- réaliser une vérification croisée des modèles du domaine et de l'application.

# Analyse de l'application (3)

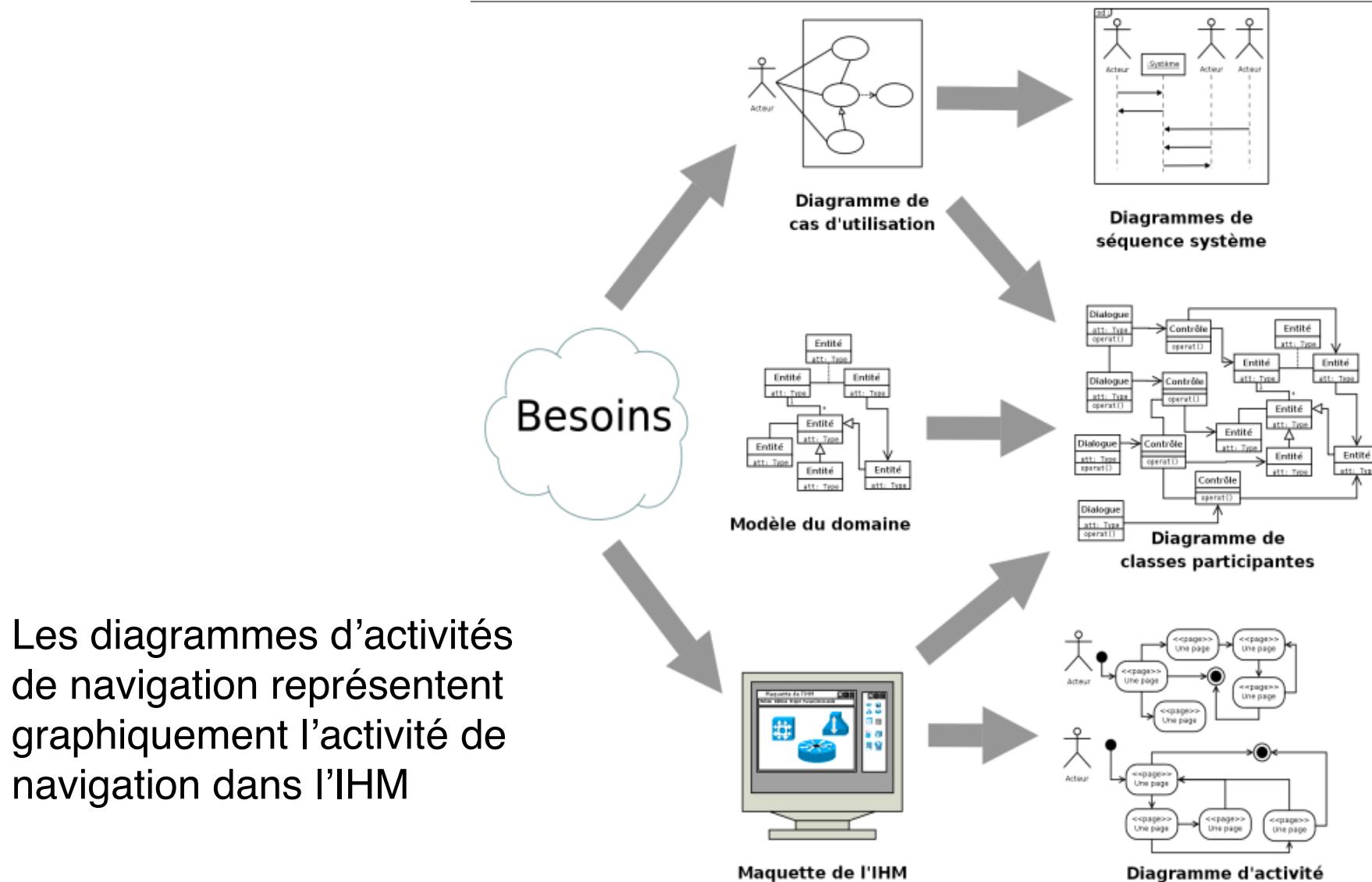
- Modèle des classes de l'application :
  - découpage en couches d'une application;
- La démarche est la suivante :
  - spécifier l'interface homme-machine ;
  - définir les classes à la périphérie du système.



## Analyse de l'application (4)

- Modèle des états de l'application
- Principe :
  - identifier des classes de l'application ayant des états complexes ;
  - trouver les événements ;
  - construire les diagrammes d'états ;
  - vérifier la cohérence avec les modèles précédents.

# L'analyse du domaine de l'application (5)

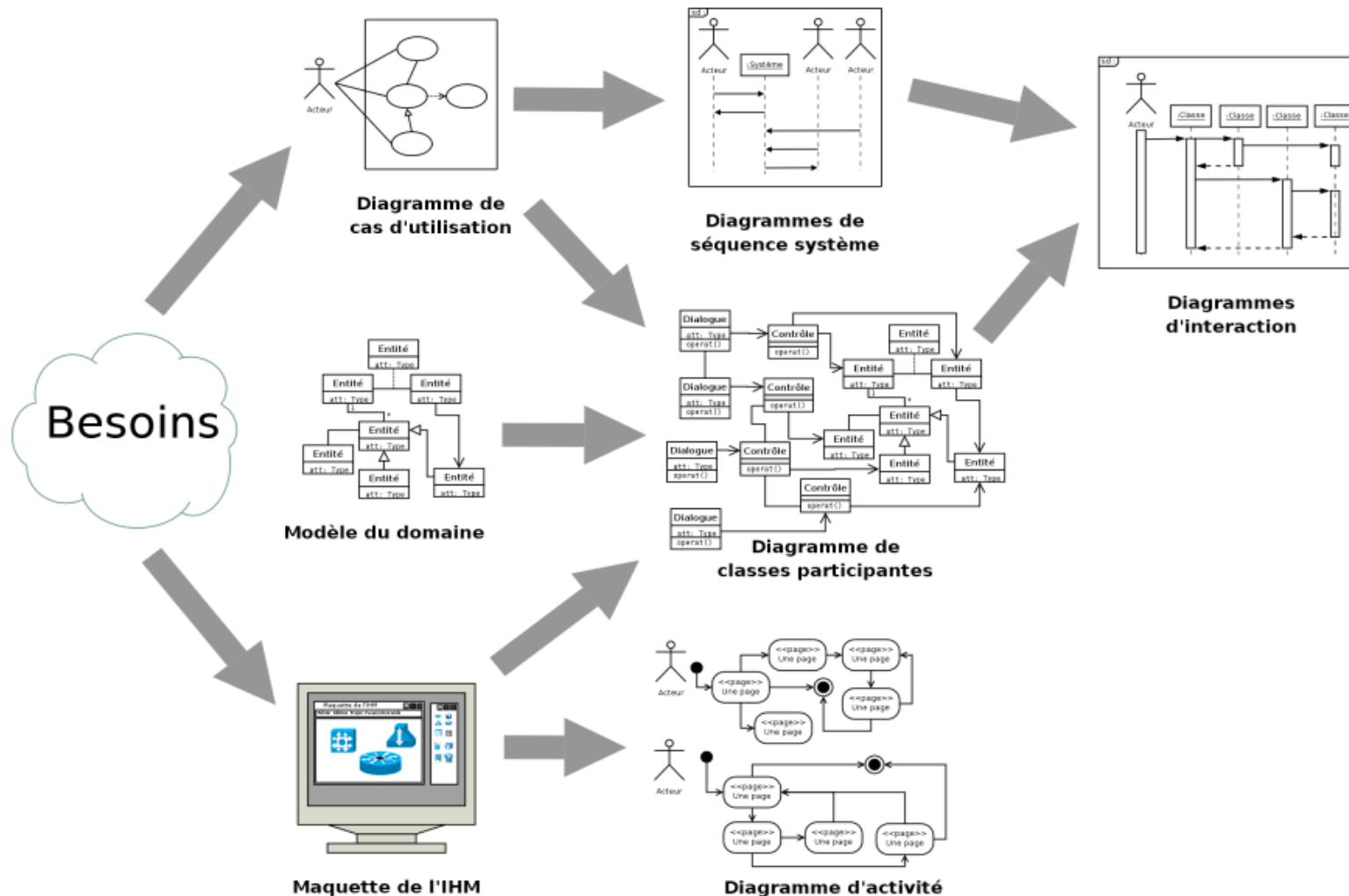


# Phase de conception (1)

- Conception du système
  - repérer les objets qui s'exécutent concurremment ;
  - allouer les sous-systèmes à des matériels ;
  - choisir sur quel(s) support(s) stocker les données (bases de données, fichiers, etc.) ;
  - identifier les ressources nécessaires (processeurs, disques, écrans, claviers, etc.) ;
  - choisir une (ou plusieurs) stratégie(s) de contrôle du logiciel (événementielle, procédurale, etc.) ;
  - s'assurer du bon fonctionnement du logiciel dans des conditions limites (comment l'initialiser, l'arrêter, gérer les erreurs, etc.).
- Conception des classes

# Phase de conception (2)

## Diagrammes d'interaction



# Phase de conception (3)

## Diagramme de classes de conception

