

Ingénierie des besoins

Chapitre II : développement d'exigences

II.3 : Analyse des exigences

2. Modélisation des exigences

Introduction

La modélisation graphique est à la fois un outil

- De recueil
- D'analyse
- De communication entre acteurs
- et de validation

Un bon schéma, vaut bien cent pages de texte (pas tj)

objectifs

- Les diagrammes peuvent être insérés dans un cahier des charges, en complément du texte écrit.
- Ils peuvent également être très utiles en tant que représentations intermédiaires(dans un but d'analyse)

Modèles en IE

- Dictionnaire de données
- Modèle E/A
- Diagramme de contexte
- Diagramme de flot de données DFD
- Table de décision
- Arbre de décision
- Diagramme état-transition
- Réseaux de Petri
- ...

Dictionnaire de données

- Un dictionnaire de donnée répertorie tous les concepts liés au domaine (as is) ainsi que ceux du futur système (to be) .
- il constitue une première base de compréhension entre les parties prenantes ;
- généralement ,un dictionnaire de données indique les infos suivantes :
 - Noms de toutes les données et leurs alias
 - Description des données
 - Structure des données (primitives ou composées)
 - Relations entre données
 - Valeurs possibles des données

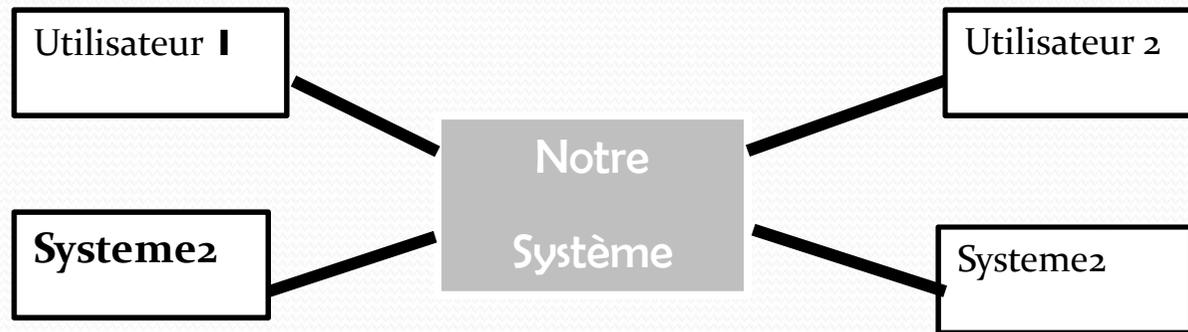
Dictionnaire de données

Exemple

donnée	Description	Observations
Livre	Toute forme d'ouvrage consultable à la biblio	Titre ,ISBN,date ...
Abonné	Toute personne inscrite à la biblio	Nom ,prénom,adresse ...
Auteur	Nom de l'auteur de l'ouvrage	1 seul auteur/ouvrage
Titre	Titre de l'ouvrage	Plusieurs exemplaire/ouvrage
Nbr exemplaire disponible	Nombre d'exemplaires disponible physiquement dans la bibliothèque	Nbr exemplaire de l'ouvrage – nbr exemplaire empruntés

Diagramme de contexte

- Le diagramme de contexte statique délimite le domaine d'étude en précisant
 - ce qui est à la charge du système et
 - en identifiant l'environnement extérieur au système étudié avec lequel il communique.



- Ses composants sont :
 - Les acteurs externes. Un acteur externe est une entité externe au système étudié qui interagit avec le système.
 - Un processus unique symbolisant le Système étudié

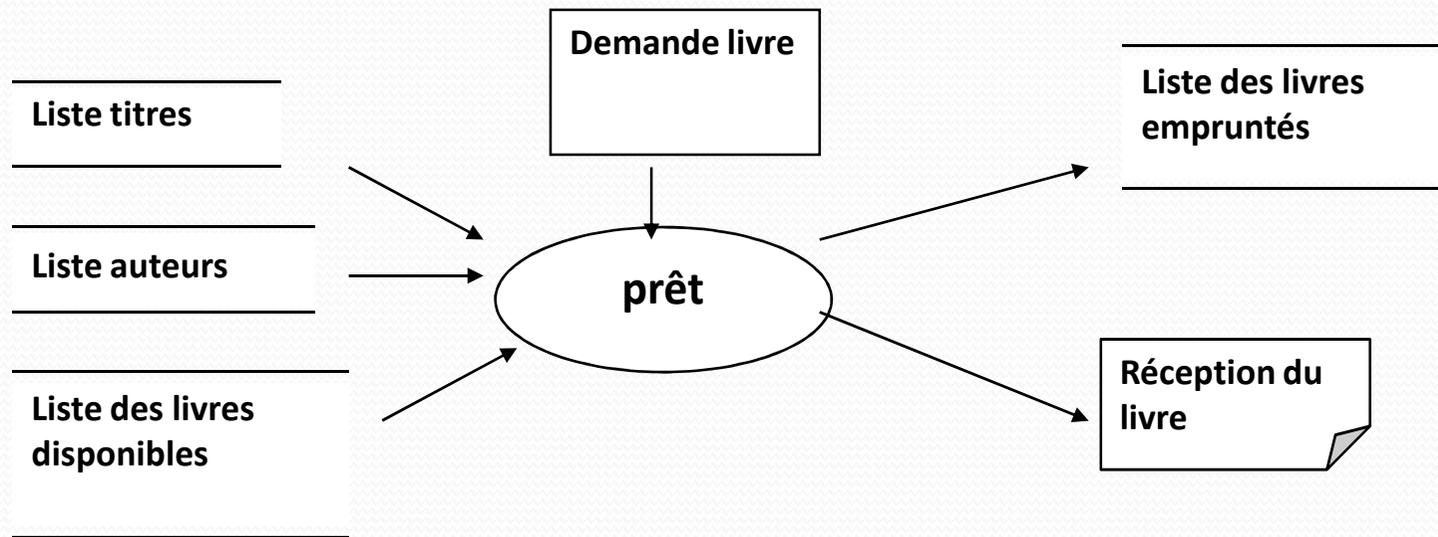
Diagramme de contexte

- La présence d'un diagramme de contexte dans une spécification traduit le fait que le rédacteur a pensé à définir le périmètre de son système. Il a identifié les interfaces à spécifier.
- A contrario son absence indique qu'il a un risque que :
 - des exigences d'interfaces soient manquantes,
 - des exigences soient hors périmètre du système

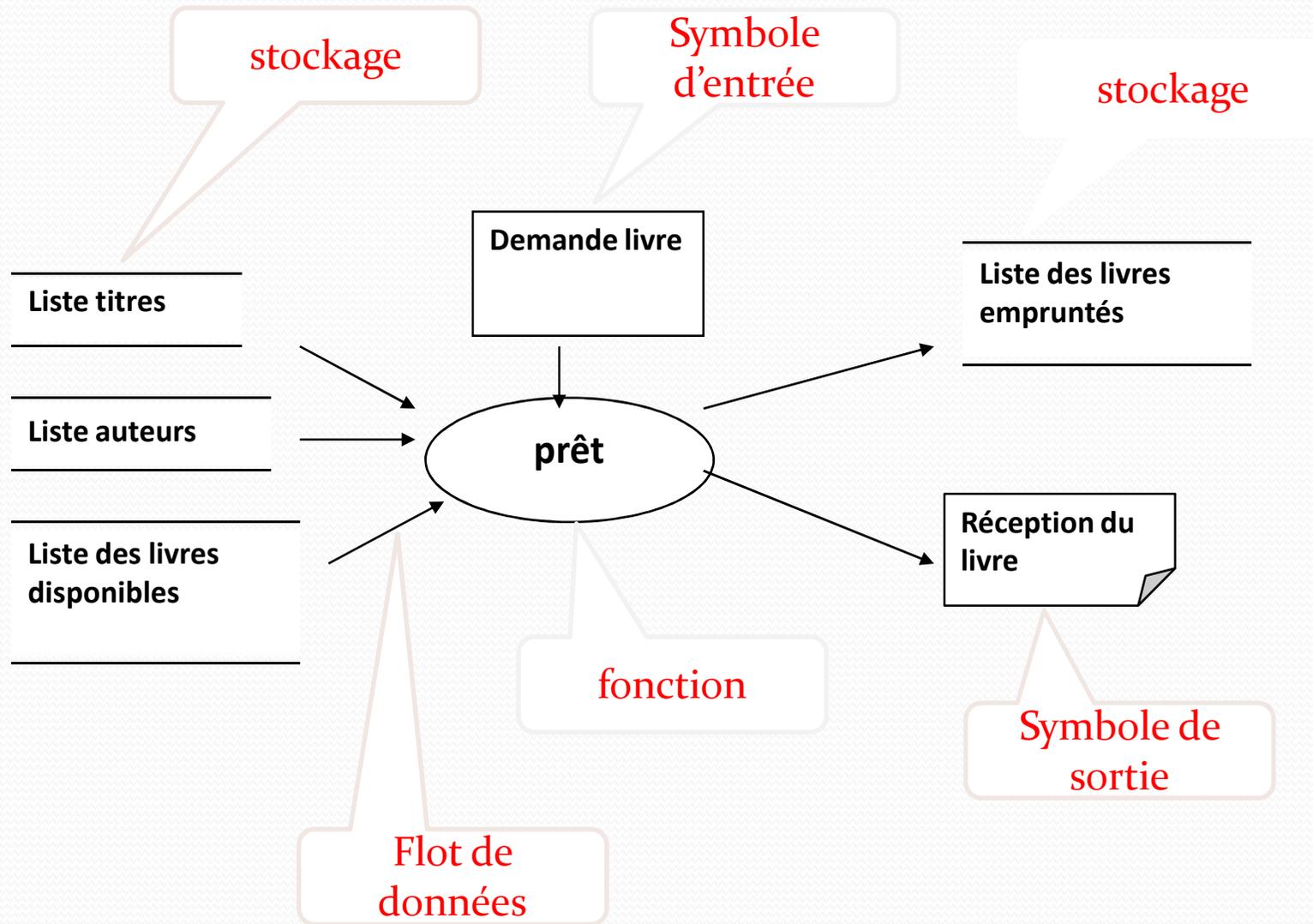
•Diagramme de flot de donnée FDD

spécifie les fonctions du système et les flots de données

- Dans des dépôts (repository)
- Entre les fonctions (flow)
- En provenance de l'extérieur (sources de données)
- En direction de l'extérieur (puits de données)



•Diagramme de flot de donnée FDD



• Diagramme de flot de donnée DFD

- Si nécessaire ,on peut appliquer des raffinements successifs sur un Dfd
- On aura ainsi plusieurs niveaux de détails
- Le raffinement s'arrête à un niveau suffisamment acceptable (généralement niveau opérationnel)
- Diagramme de contexte est le niveau 0 dans le DFD

Table de décision

Une table de décision exprime exhaustivement les relations entre la logique interne d'une fonction ou d'un processus, les données fournies en entrée et les données restituées en sortie.

- Les données fournies en entrée deviennent les conditions de la table de décision.
- Les données restituées en sortie proviennent des actions déclenchées par la table de décision.
- Les couples (conditions, actions) deviennent les règles de la table de décision .)

- Table de décision

Nom de la table	Règle 1	Règle2	Règle l
Condition 1	Oui		Non
Condition 2	Oui		Oui	...
....				
Condition n	Non		Oui	...
Action1	x			
Ation2				
....				
Action m				

- Les tables de décision apportent une méthode claire pour vérifier les tests de toutes les combinaisons de conditions possibles et pour vérifier que toutes les combinaisons possibles sont gérées par le logiciel testé.

Table de décision

exemple : processus de vente

Vente	R1	R1	R2	R3	R4	R5
Client fidèle ou nouveau	N	O	O	O	O	O
Temps suffisant	-	-	O	O	N	N
Quantité disponible	-	O	N	N	O	N
Main d'œuvre et matière première suffisante	-	-	O	N	-	-
Fabriquer			X			
Acheter				X		X
Livrer		X	X	X	X	X
Refuser commande	X					

Table de décision

limitations/difficultés/risques

- La plus grande difficulté de cette technique est d'identifier toutes les conditions et d'en définir le résultat
- La combinatoire peut vite augmenter et il devient humainement difficile de poser la table de décision dans sa totalité

Couverture

le nombre de cas de test pour avoir 100% de couverture est égal à $2^{\text{nombre de conditions}}$

Table de décision

Mise en œuvre

- Comme pour toute technique basée sur les spécifications, la première étape est d'identifier les conditions présentes dans cette dernière mais aussi les actions du logiciel en sortie.
- Une fois les conditions identifiés, nous posons la table de décision (comme pour une table de vérité).

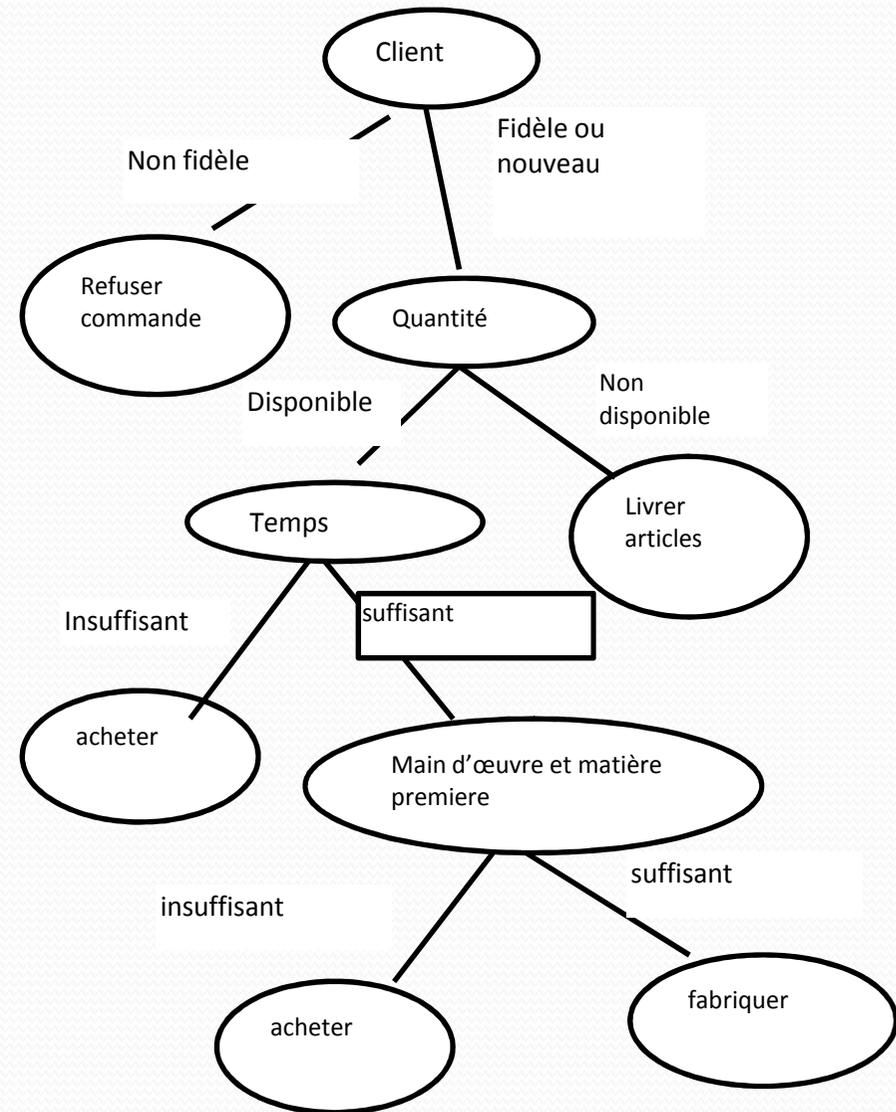
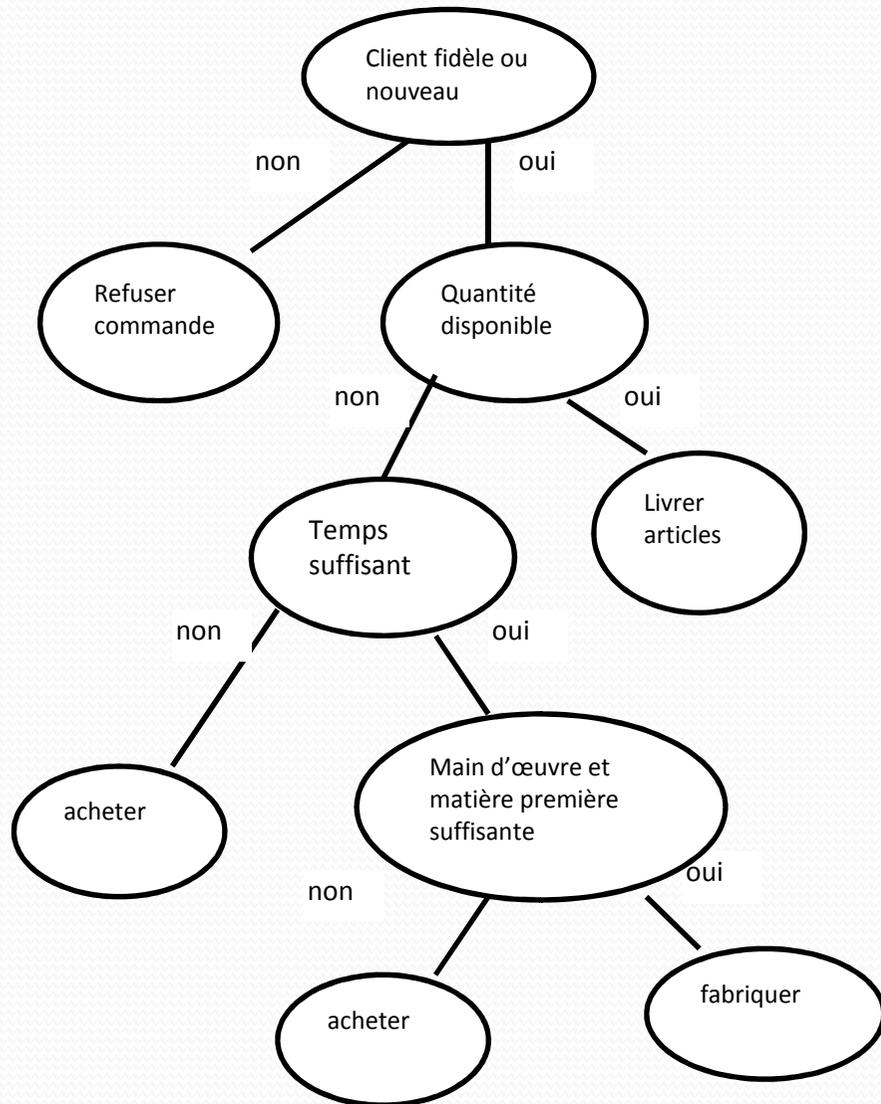
Arbre de décision

Arbre où :

- chaque noeud interne porte un test booléen sur un attribut
- chaque branche correspond à un résultat du test (vrai ou faux),
- chaque feuille est étiquetée par une classe.(action de décision)

Modeles en IE

• Arbre de décision



- Diagramme d'activité (carte processus:process map

- Il permet de représenter les étapes d'un processus métier, leur séquence, leurs entrées et sorties, les acteurs responsables de chaque étape.
- Ces diagrammes peuvent être utilisés pour représenter un processus existant ou à venir, automatisé ou manuel.
- Les étapes peuvent être réparties sur des « bandes » (en anglais *swim lanes*, par analogie avec les couloirs d'une piscine), chaque bande étant sous la responsabilité d'un acteur

Diagramme d'activité

Couloir de responsabilité

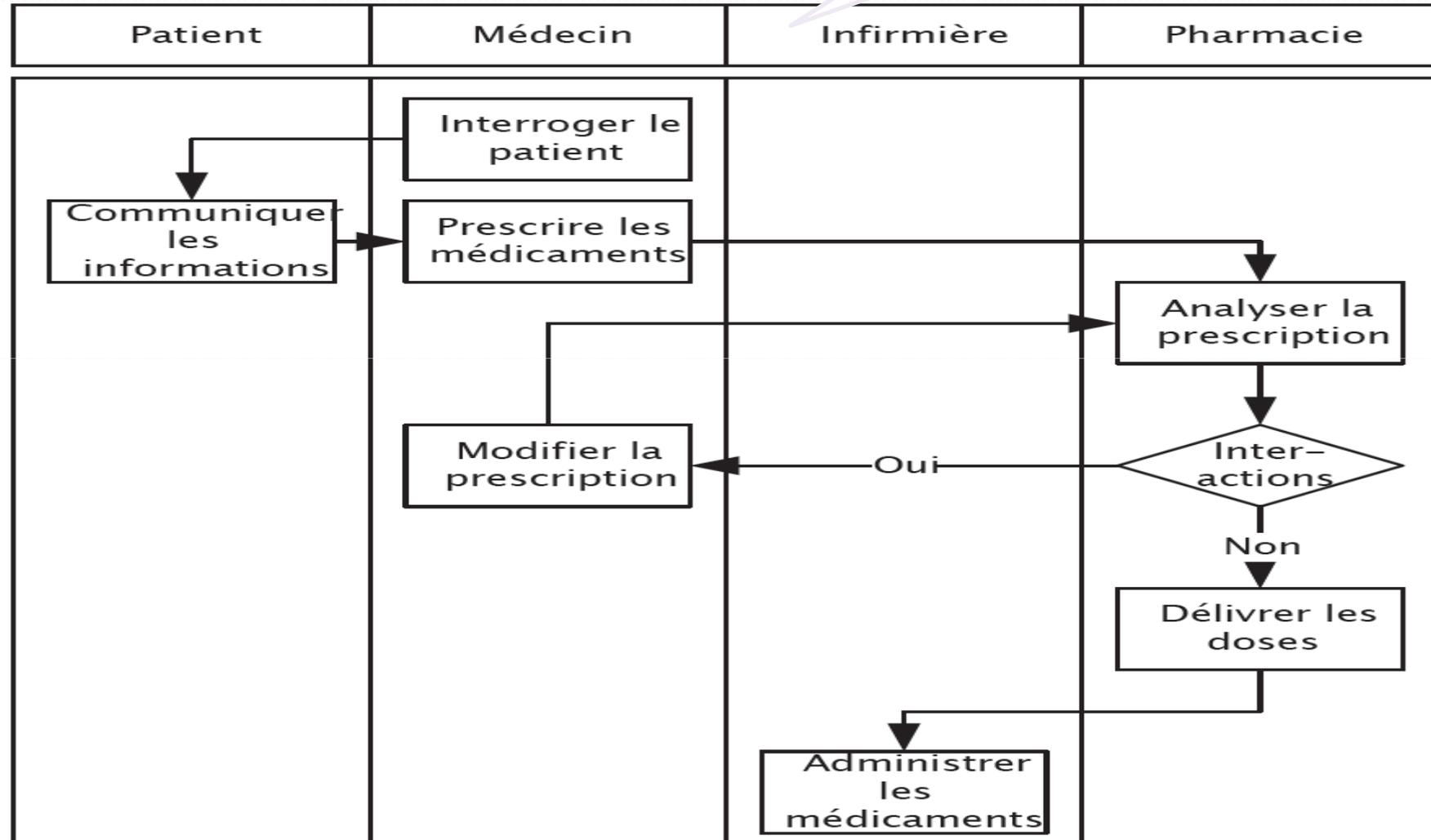


Diagramme de séquence

- permet de modéliser les flux d'informations entre acteurs, en faisant ressortir l'enchaînement chronologique des activités d'un processus.
- Le diagramme de séquence est un outil d'analyse plus que de recueil ou de spécification. En dehors des domaines techniques où la synchronisation des tâches est complexe
- utile pour alimenter la réflexion permettant de décider des opérations à automatiser et sur les processus à optimiser à la suite d'une informatisation

Diagramme de séquence

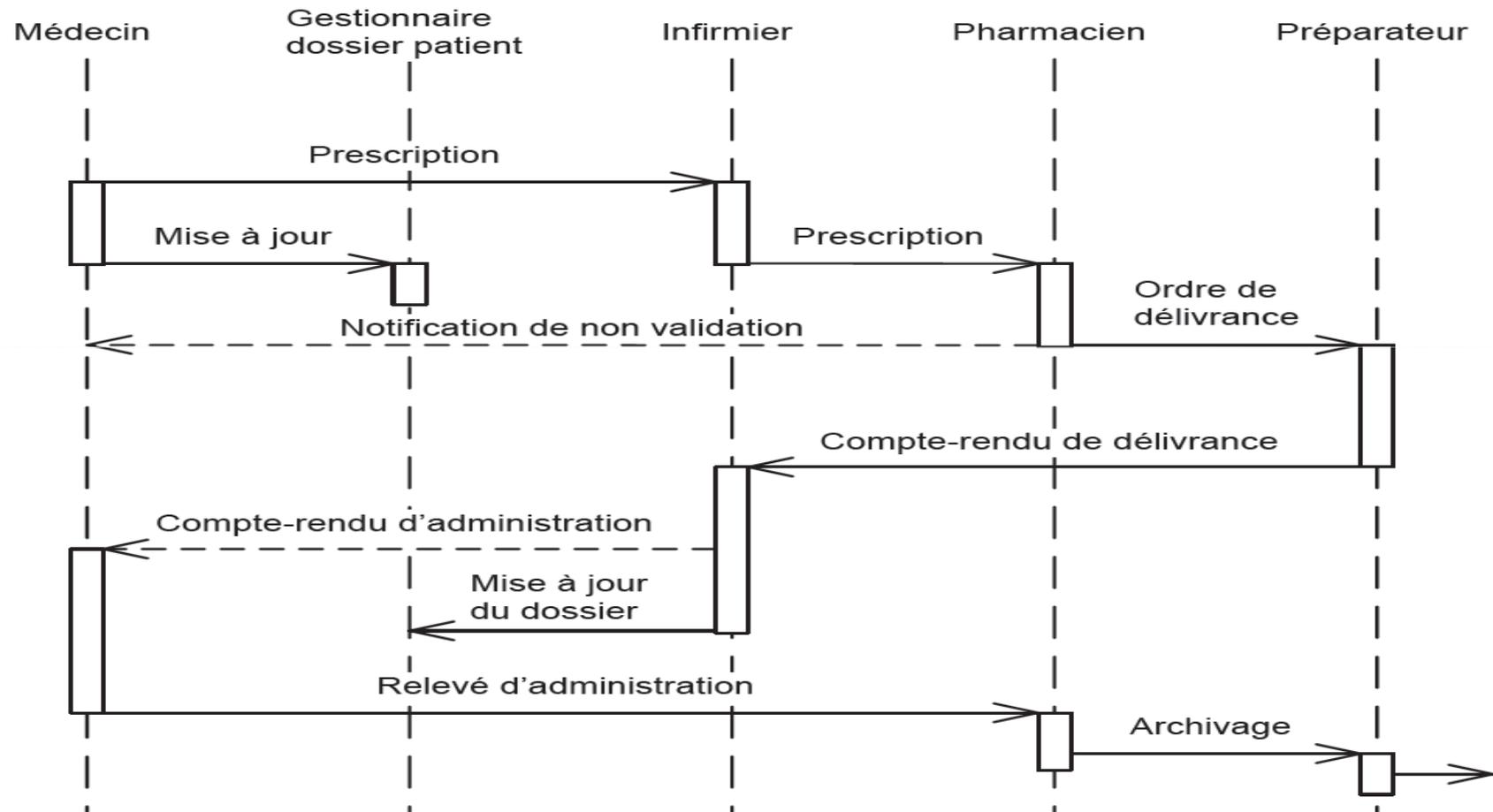


Diagramme Entite/association ,de classes

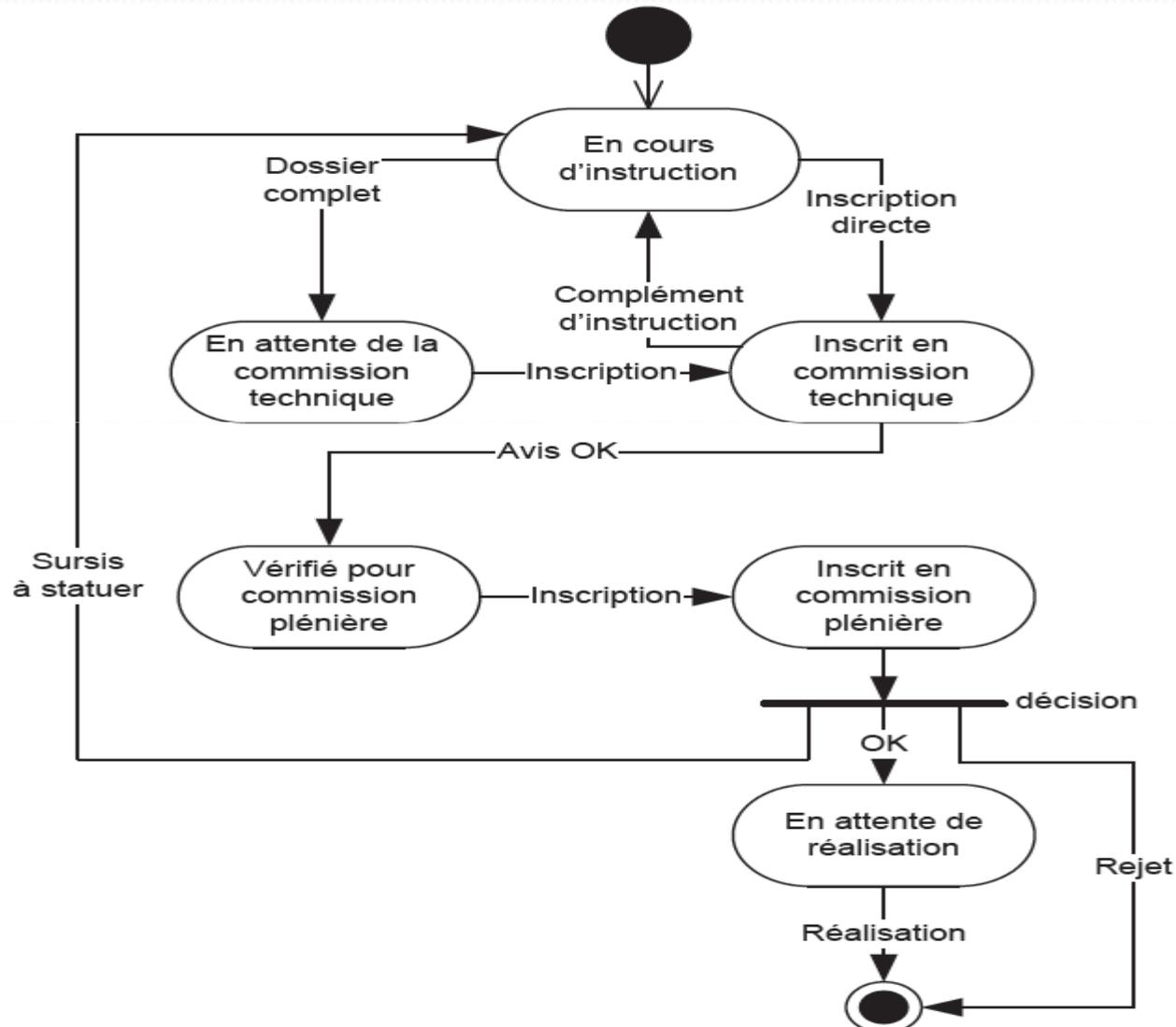
- Permet de représenter les attributs des données et les relations entre les données du domaine.
- permet d'avoir une vision exhaustive des données manipulées par le système.
- Représenter quelques règles de gestion du domaine(entités , cardinalité, identifiant ...)

Diagramme états-transitions

- Permet de décrire la séquence des étapes d'un processus.
- Décrire la séquence des états d'un système , sous systèmes , objet du domaine

•Diagramme états-transitions

•Exemple : projet



Cas d'utilisation

- Les cas d'utilisation sont un excellent outil d'analyse, mais aussi de recueil et de spécification.
- utiles pour assembler les besoins recueillis auprès de plusieurs acteurs, travailler avec les différents acteurs, interagir avec eux, et faire valider les exigences correspondantes.
- En d'autres termes, c'est un bon moyen de transformer un besoin en exigence.

Maquettes

- Une maquette est un modèle qui préfigure un produit futur sous forme de dessin sur papier ou électronique (mage)
- Une maquette est toujours jetable .
- Le but n'est pas de commencer à développer le produit, ni de spécifier.
- Permet de visualiser les fonctions du futur produit (futur use cas)

prototypes

- Contrairement à la maquette, un prototype est « vivant ». C'est du logiciel.
- Il peut être jetable ou au contraire réutilisable, si on a l'intention de l'intégrer dans le produit fini.
- Prototyper, c'est donc entrer provisoirement (et par exception) dans la phase de réalisation pour expérimenter un fragment du produit à venir
- le prototypage est faisable si une équipe de développement est présente et interagit avec la maîtrise d'ouvrage!

prototypes

Types :

1. Le prototype « horizontal » : il simule l'application visuellement (enchaînement d'écrans), sans que toutes les fonctions soient présentes.
1. Le prototype « vertical » : une seule fonction est complètement développée. Ce type de prototype est utilisé pour tester une fonction ou une propriété du logiciel, comme la fiabilité, la sécurité, ou le temps de réponse.

Conclusion

- L'objectif premier est de se faire comprendre de toutes les parties prenantes et d'ajouter de la précision et lever les ambiguïtés .
- Pour représenter la même information, on a parfois le choix entre plusieurs outils. Inversement, un outil peut avoir plusieurs usages.
- Il ne s'agit pas de « tout » représenter avec le même outil, ni d'utiliser tous les outils à tout prix