

Spécification et gestion des besoins

Le cahier de charges et spécification du logiciel

Le contenu est basé sur les transparents de la 10^{ème} édition de “*Software Engineering*” de Ian Sommerville

Le besoin(l'exigence)

- Les besoin ont un service dual.
 - Base d'une offre – ouverts et objet de négociation
 - Base du contrat – spécifications détaillées
- Besoins d'utilisateurs (pour les utilisateurs, définitions)
- Besoin du système (pour les développeurs, spécifications)
- Spécification système

Définitions et spécifications

- Exemple LIBSYS

- Définition

1. Le logiciel doit mettre à disposition des moyens d'accès et de présentation des fichiers externes produits par autres logiciels.

- Spécification

- 1.1 Des moyens pour déterminer le type des fichiers externes doit être disponible
- 1.2 Pour chaque type de fichier externe on doit avoir un outil associé pour traiter le fichier
- 1.3. Chaque type de fichier externe est présenté comme une icône spécifique sur l'écran.
- 1.4. L'utilisateur doit avoir la possibilité de définir ces propres icônes.
- 1.5. Quand une icône est choisie l'outil associé est appliqué.

Types de besoins

- Types de besoins
 - Fonctionnels
 - Non-fonctionnels – contraintes
 - Techniques
 - De l'environnement
 - De l'utilisateur
- Du domaine – venus du domaine d'application.
Ils peuvent être fonctionnels et non-fonctionnels.

Besoins fonctionnels

- Ils décrivent la fonctionnalité ou les services du système
- Exemple – LYBSYS système

Systeme bibliothécaire qui assure un interface unique pour un grand nombre bases de données d'articles qui ce trouvent en différentes bibliothèques.

Les utilisateurs cherchent, téléchargent et impriment les article pour leur études personnels

Exemples des BF

- Exemples

Utilisateur doit pouvoir chercher dans l'ensemble initial de bases ou dans un sous-ensemble

Le système doit assurer un dispositif approprié pour que l'utilisateur soit capable de lire les documents.

A chaque commande est affecté un nombre (ORDER_ID) qui peut être copié dans l'espace du permanent de la compte

- Les besoins peuvent être compris différemment par les utilisateurs et développeurs

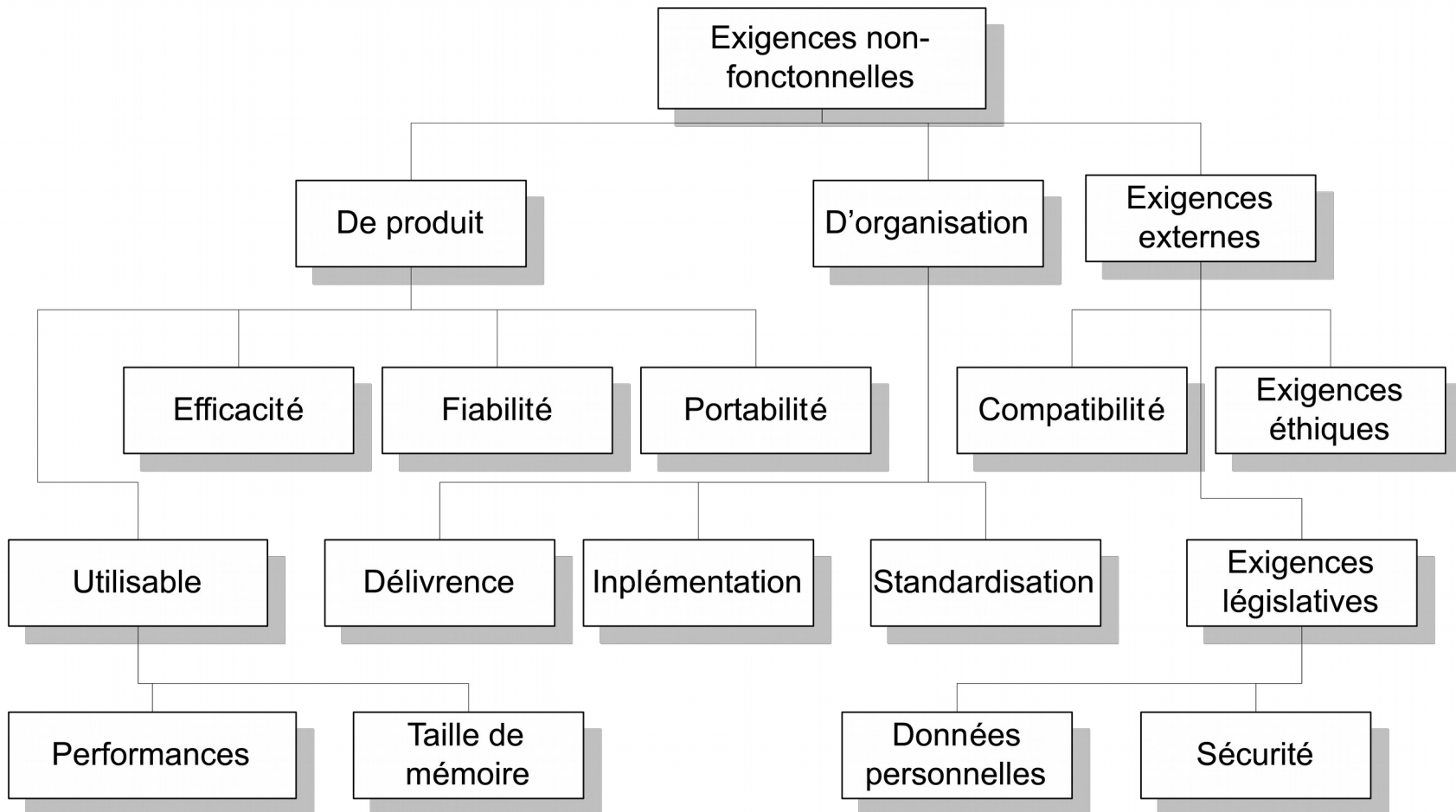
Propriétés souhaitées des BF

- Complets
- Cohérents – sans contradictoires
- En fait c'est presque impossible

Besoins non-fonctionnels

- Qu'est ce qu'ils présentent?
 - Propriétés du système
 - Fiabilité
 - Temps de réaction
 - Taille de mémoire nécessaire
 - Contraintes
 - La vitesse des unités d'entrée/sortie
 - Les outils CASE utilisés
 - La résolution d'écran
- Importance – Même plus grande que les besoins fonctionnels

Besoins non-fonctionnelles



Besoins non-fonctionnels

- Exemples

- Besoins de produit

8.1. L'interface d'utilisateurs doit être implémentée en simple HTML sans cadres et applets Java

- Besoins organisationnels

9.3.2 Le processus de développement et la documentation doit conformer avec le standard XYZCo-SP-STAN-95

- Besoins externes

7.6.5. Le système ne doit exposer aux opérateurs du système aucune information personnelle des clients sauf leur numéro personnel et leur nom

Mesure des besoins NF

- Buts – intention générale de l'utilisateur

Le système doit être facilement utiliser par des contrôleurs expérimentés et il doit organisé de telle façon que le nombre d'erreur soit minimisé.

- Besoin non-fonctionnel vérifiable

Contrôleurs expérimentés doivent être capables d'utiliser le système après 2 heures de formation. Après cette formation le nombre moyen d'erreurs faites par utilisateurs expérimentés ne doit pas passer 2 par jour.

Mesure des besoins NF

Propriété	Mesure
Vitesse	Nombre de transactions par seconde Temps de réaction Temps de rafraîchir l'écran
Taille	Kbytes, Mbytes, GBytes
Facile pour utiliser	Temps pour formation Nombre d'écrans^pages d'aide
Fiabilité	Temps moyen sans panne Probabilité de refus d'accès Fréquence d'apparition d'erreurs Disponibilité
Robustesse	Temps de restart après une panne. Pourcentage des événements causant une panne Probabilité de perte ou corruption des données
Portabilité	Pourcentage des instructions qui son machine dépendantes. Nombre de systèmes cibles

Contradiction des besoins

- Les besoins se contredisent les uns les autres très souvent
- Exemple
 - des puces dans un vaisseau d'espace
 - des Applications Mobiles sans connexion
 - Sécurité et Rapidité
 - Voiture de course blindée....

Besoins du domaine

- Ils sont dérivés du domaine du système
- Importance – très haute
- Difficultés
 - Ils sont rédigés dans la langue du domaine et ne sont pas compris très bien par les développeurs.
 - Pour les experts du domaine ils sont implicites (par défaut)

Besoins du domaine

- Exemples

- LYBSYS

Il y aura interface standard vers toute base de donnée basée sur le standard Z39.50

Par raison de contrainte du code de la propriété intellectuelle, les documents doivent être supprimés immédiatement dès leur venue. Selon les besoins d'utilisateur ces documents doivent être imprimés sur le serveur local ou être dirigé vers une imprimante de réseau.

- Système de protection de train

La décélération d'un train est calculée comme:

$$D_{\text{train}} = D_{\text{control}} + D_{\text{gradient}}$$

où D_{gradient} est $9.81\text{m/s}^2 * \text{gradient compensé}/\alpha$ et où les valeurs du gradient compensé/ α sont connues pour les différents types de trains

Besoins d'utilisateur

- Exprimés dans une langue naturelle, tableaux et diagrammes et ils doivent être compris par tout le monde.
- Dangers
 - Manque de clarté
 - Confusion - mixte de besoins fonctionnels et non-fonctionnels
 - Amalgamation – Union de plusieurs besoins.

Exemples

- LYBSYS

4.5 LYBSYS doit assurer un système de comptabilité qui doit maintenir les paiements d'utilisateurs. Les directeurs peuvent configurer le système de telle façon qu'un utilisateur puisse recevoir un rabat.

- Problème

- Mixte d'information conceptuel et détaillé

Exemples

- Quadrillage d'éditeur graphique

2.6 Moyens de quadrillage – pour assister au positionnement d'éléments dans un diagramme l'utilisateur peut activer un quadrillage en centimètres ou en pouces, par une option du menu. Au début le quadrillage est inactif. Il peut être activé ou désactivé n'importe quand et il peut passer de pouces en centimètres et vice-versa. Quand la taille du diagramme est petite cette option peut être activée mais avec un nombre de lignes réduit afin d'éviter l'encombrement du diagramme.

- Problème - 3 différents types de besoins
 - Besoin fonctionnel (l'activation du quadrillage)
 - Besoin non-fonctionnel (les unités du quadrillage)
 - Besoin d'interface d'utilisateur (commutation d'unités de mesure)

Recommandations

- Développez un format standard et l'utiliser après
- Utilisez la langue d'une manière constante.
Utilisez futur pour les besoins obligatoires et « il sera bien que » pour les besoins optionnels.
- Soulignez le texte afin d'identifier les plus importantes parties du besoin
- Evitez l'usage d'argot informatique

Les besoins détaillés (du système)

- Particularités
 - Des spécifications plus détaillées
 - Base de la conception du système
 - Ils peuvent être inclus dans le cahier de charge et dans le contrat
 - On peut utiliser et des modèles du système (les UML diagrammes)
 - Ils sont interdépendants avec l'architecture du système

Langue de description de besoins

- Désavantages de la langue naturelle
 - Ambiguïté
 - Sur-flexibilité
 - Manque de structure
- Alternatives
 - Langues structurées – formulaires, modèles, tableaux
 - Langages spécifiques – rarement utilisés
 - Notations graphiques – UML
 - Spécifications mathématiques

Formulaire

- Formulaire standard (exemple)
 - Définition de la fonction.
 - Description de saisie et son origine.
 - Description des sorties and et leurs destinations.
 - Indication d'autres entités dont on a besoin.
 - Pré and post conditions (s'il y en a).
 - The effets secondaires (s'il y en a) de la fonction

Formulaire - Exemple

Insulin Pump/Control Software/SRS/3.3.2

Function Compute insulin dose: Safe sugar level

Description Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.

Inputs Current sugar reading (r2), the previous two readings (r0 and r1)

Source Current sugar reading from sensor. Other readings from memory.

Outputs CompDose – the dose in insulin to be delivered

Destination Main control loop

Action: CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.

Requires Two previous readings so that the rate of change of sugar level can be computed.

Pre-condition The insulin reservoir contains at least the maximum allowed single dose of insulin..

Post-condition r0 is replaced by r1 then r1 is replaced by r2

Side-effects None

Spécification tabulaire

- Utilisé comme un complément de la langue naturelle
- Bonne pour définir plusieurs activités alternatives .

Spécification tabulaire

Condition	Action
Sugar level falling ($r2 < r1$)	CompDose = 0
Sugar level stable ($r2 = r1$)	CompDose = 0
Sugar level increasing and rate of increase decreasing ($(r2-r1) < (r1-r0)$)	CompDose = 0
Sugar level increasing and rate of increase stable or increasing. ($(r2-r1) \geq (r1-r0)$)	CompDose = round $((r2-r1)/4)$ If rounded result = 0 then CompDose = MinimumDose

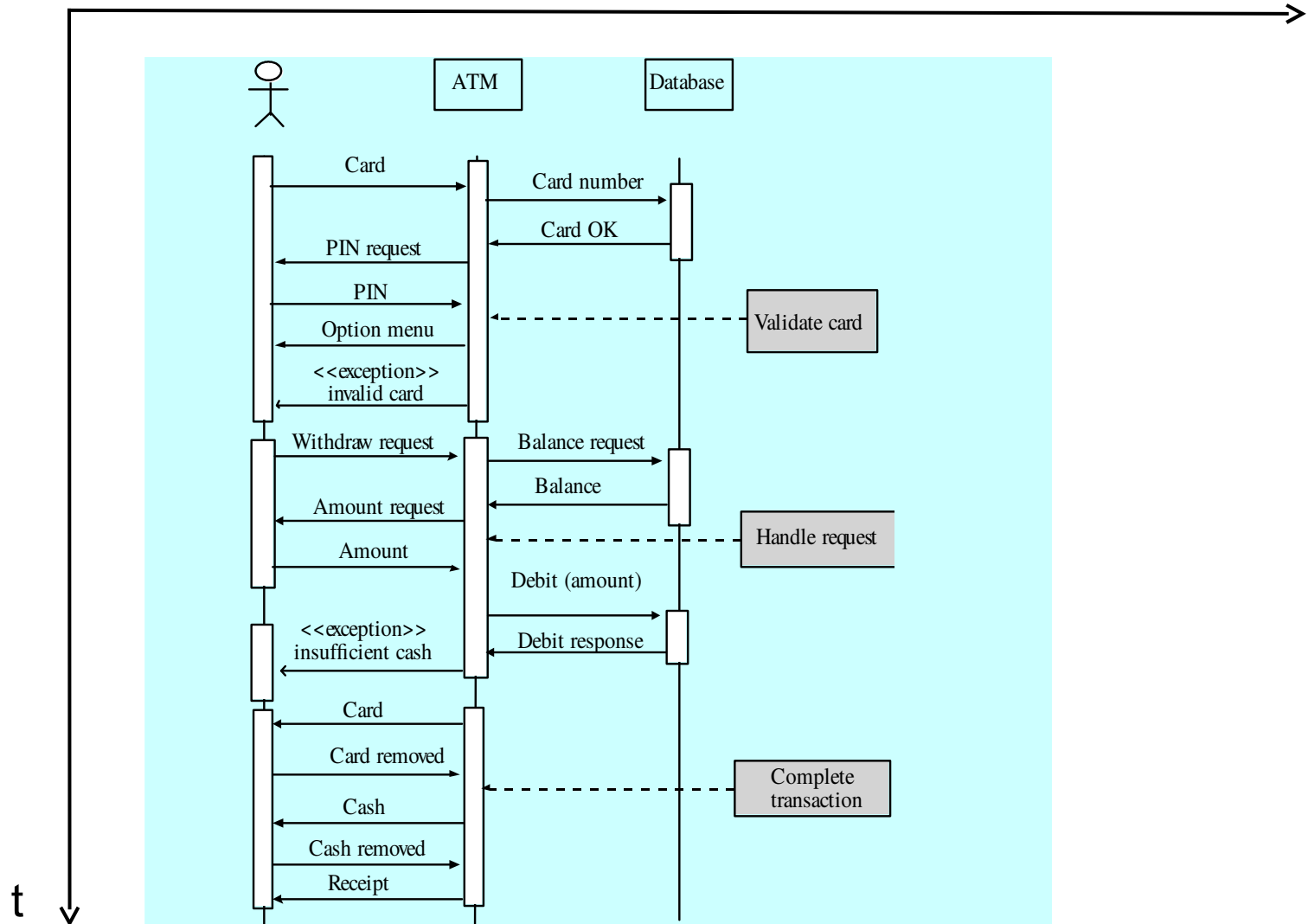
Modèles graphiques

- Très bons pour présenter différents points de vue. Ils sont utilisés pour présenter des structures, des changements des états ou une séquence d'activités.

Diagrammes de séquences

- Ils présentent des séquences d'événements produits par l'interaction d'un utilisateur avec le système.
- Tirer des billets d'un DB (ATM)
 - Valider la carte
 - Traiter la demande
 - Effectuer la transaction.

Diagrammes de séquences Exemple



Spécification d'interfaces

- Spécification d'interfaces avec autres systèmes
- Types
 - Procédural
 - Les structures des données échangées
 - La présentation des données
- Langage – une notation formalisée

```
interface PrintServer {  
  // defines an abstract printer server  
  // requires: interface Printer, interface PrintDoc  
  // provides: initialize, print, displayPrintQueue, cancelPrintJob, switchPrinter  
  void initialize ( Printer p ) ;  
  void print ( Printer p, PrintDoc d ) ;  
  void displayPrintQueue();  
}
```

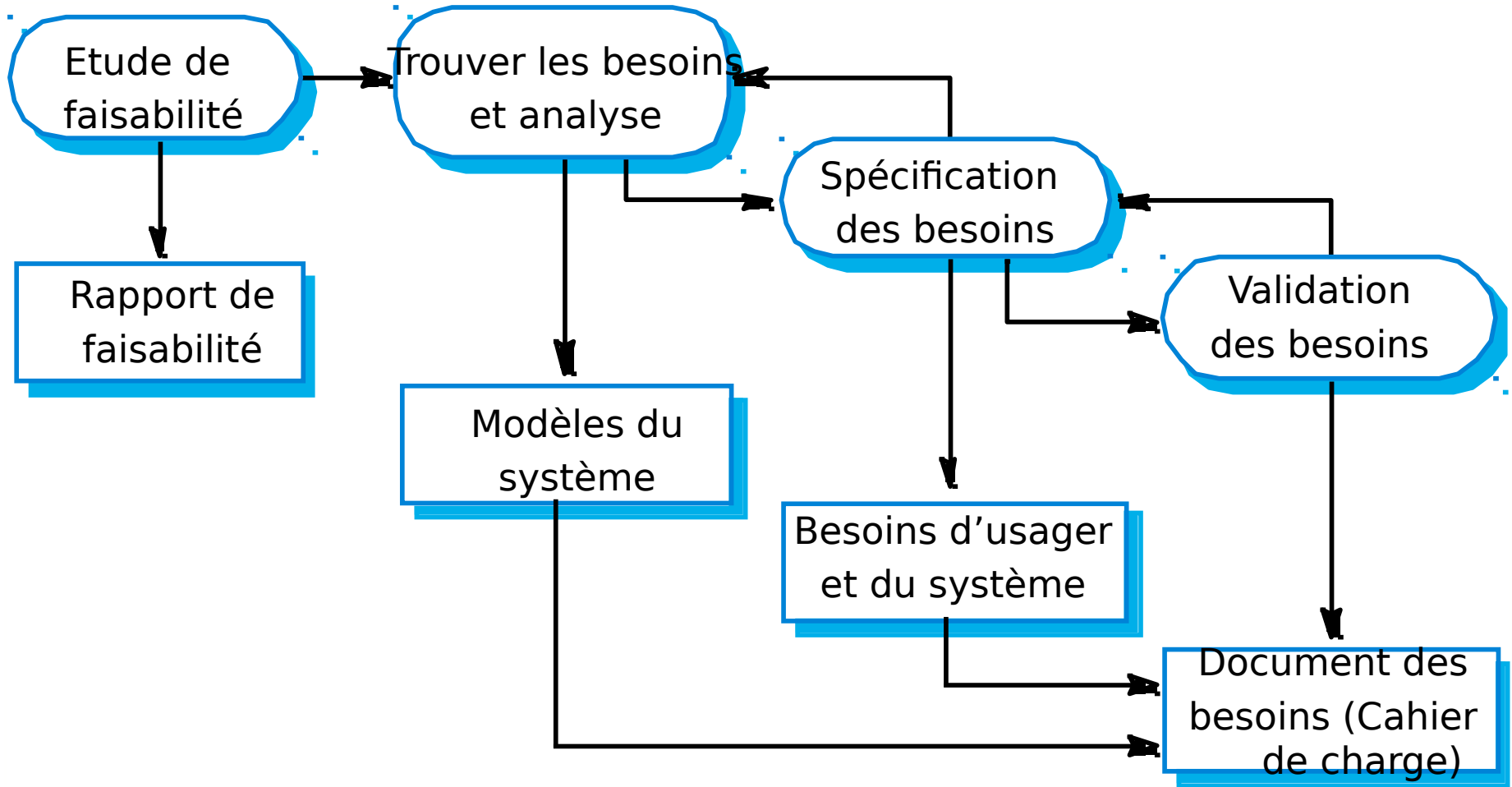
Le cahier de charges

- C'est le document principal. Il doit contenir
 - Les besoins d'utilisateur
 - Les spécifications des besoins
- Objectif
 - Base de contrat
 - Base de conception et développement
- IEEE structure standard
 - Introduction.
 - Description générale.
 - Besoins spécifiques.
 - Appendices.
 - Index.

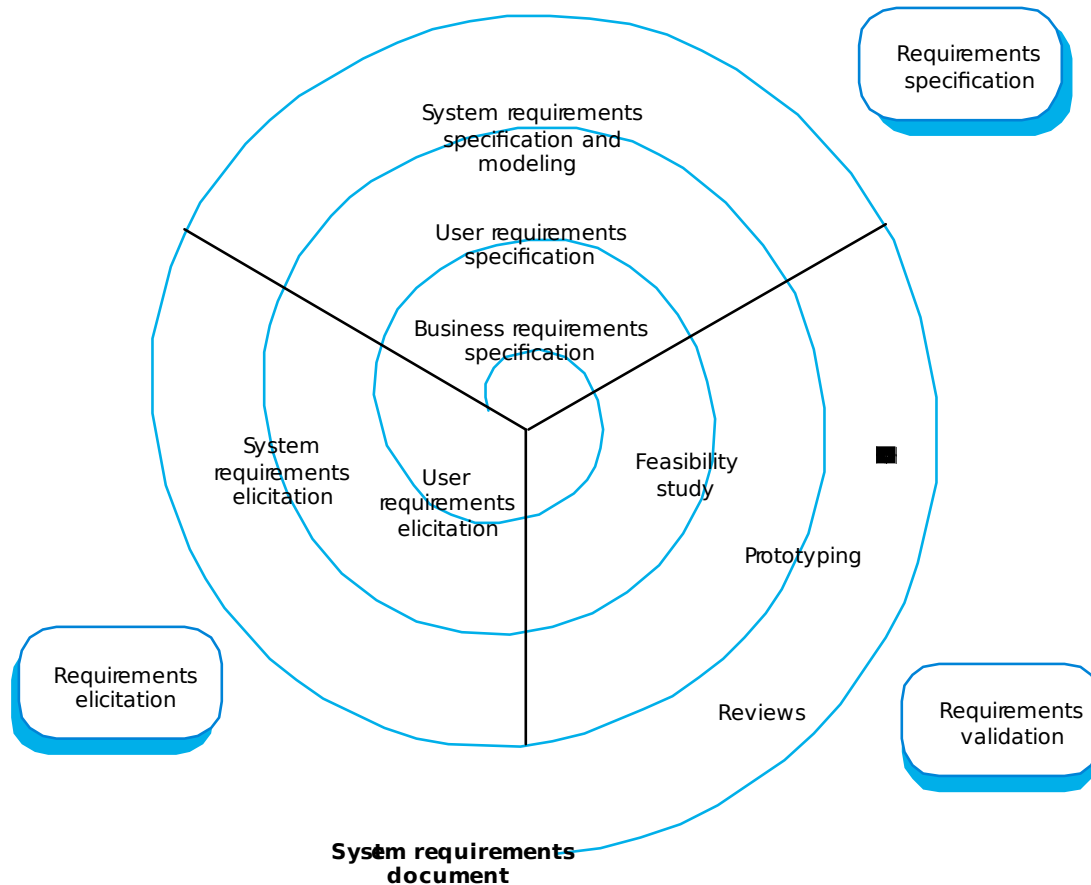
Structure générale

- Préface
- Introduction
- Glossaire
- Définition des besoins d'utilisateur
- Architecture du système
- Spécification des besoins système
- Modèles du système
- Evolution du système
- Appendices
- Index

L'ingénierie des besoins



Le processus spirale



L'étude de faisabilité

- Une étude courte durée qui vérifie si le système proposé vaut la peine.
- Objectifs - Vérifie si:
 - Le système va améliorer l'organisation
 - Il peut être fait avec la technologie contemporaine et avec le budget donné
 - Il peut être intégré avec les autres systèmes existants

Découverte et analyse

- Le monde engagé - actionnaires
 - Le personnel technique qui travaille avec les clients
 - Le domaine
 - Les contraintes opérationnelles
 - Managers
 - Utilisateurs
 - Ingénieurs de maintenance
 - Expertes dans le domaine
 - Unions professionnels
 - Les organismes communales et gouvernementales

Problèmes

- Les actionnaires ne savent pas qu'est ce qu'ils veulent
- Ils expriment les besoin en leur propres termes.
- Les différents actionnaires ont des besoins contredisants.
- Des facteurs politiques et organisationnels peuvent influencer les besoins.
- Les besoins et les actionnaires aussi peuvent changer pendant le durée du processus

Activités

- Découverte
- Classification et organisation
- Définir les priorités et négociation – résoudre les contradictions
- Faire la documentation

Découverte des besoins

- Information collectée
 - Concernant les systèmes proposés et existants
 - En distillant on peut obtenir les besoins d'utilisateur et du système
- Sources
 - Documentation
 - Actionnaires
 - Spécifications des systèmes similaires

Exemple

- Les actionnaires du système de distributeurs de billets (ATM)
 - Clients de la banque
 - Représentatives d'autres banques
 - Directeurs de la banque
 - Le personnels des guichets
 - Administrateurs de BD
 - Chefs de sécurité
 - Le département de marketing
 - Ingénieurs de maintenance de matériel et logiciel
 - Régulateurs des banques

Points de vue

- Points de vue des acteurs
- Points de vue indirectes
- Points de vue du domaine

Interviewer

- Types d'interviews
 - Fermé
 - Ouvert
- En pratique
 - Mixte d'interviews ouverts et fermés
 - Bons pour comprendre les expectations des actionnaires
 - Mauvais pour les besoins du domaine
- Interview effectif
 - Jamais « Qu'est ce que voulez-vous du système »
 - Propositions et suggestions

Scénarios

- Des exemples de la vie réelle qui montrent comment le système peut être utilisé
- Structure
 - Description de la situation au début
 - Description du flux normal des événements;
 - Description des choses qui peuvent échouer;
 - Information pour des activité simultanées;
 - Description de la situation à la fin du scénario;

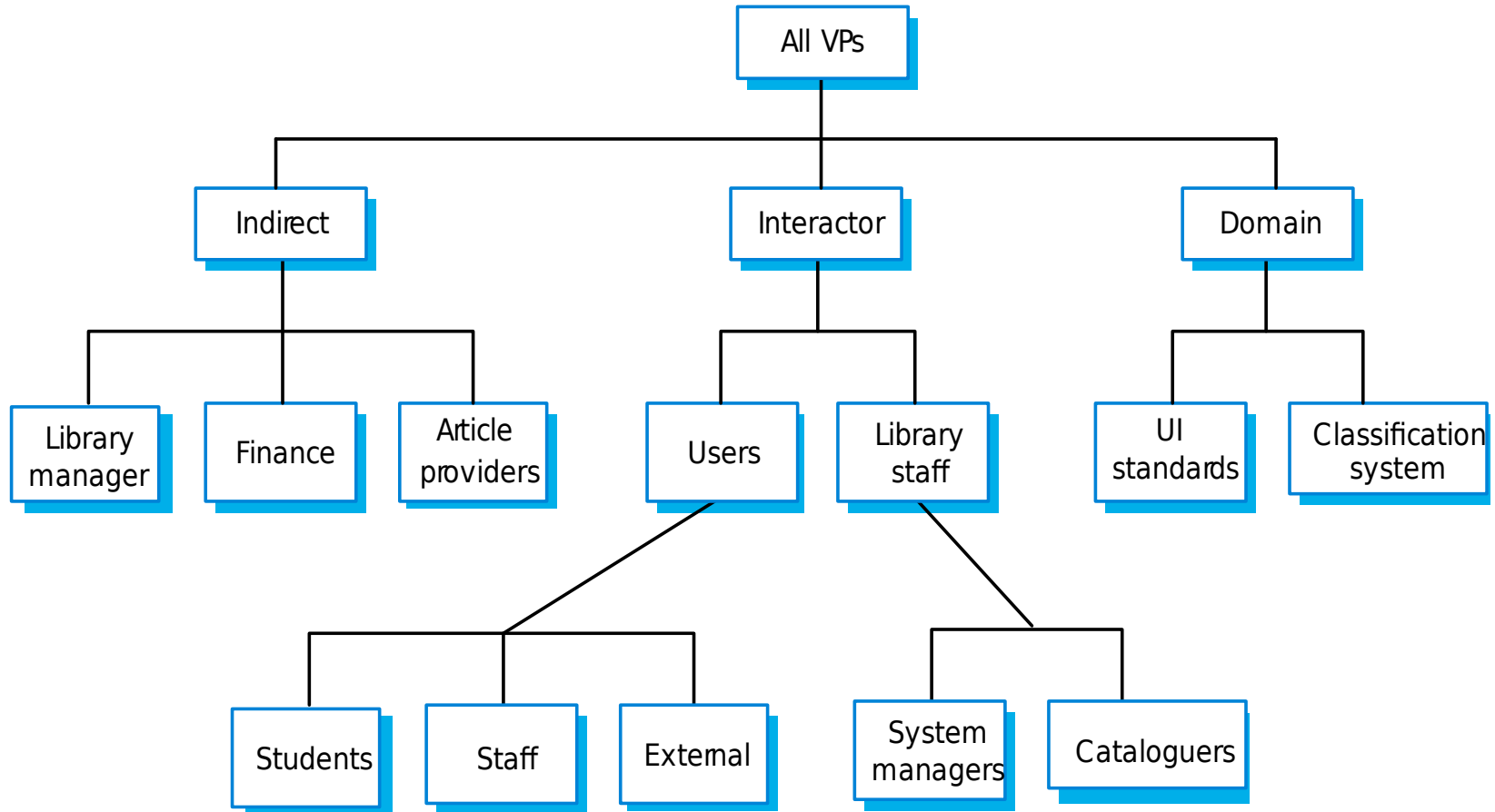
Validation des besoins

- Démontrer que les besoins définissent le système désiré par le client.
- Les erreurs sont trop chères.
- Propriétés qui doivent être vérifiées
 - Validité – les fonctions du système sont elles adaptées aux besoins du client?
 - Consistance – Y a-t-il des confits entres les besoins?
 - Totalité – Toutes les fonctions sont présentes?
 - Réalisable – Est-ce qu'on peut les développer?
 - Vérifiable – Peut-on les vérifier?

Techniques de validation

- Revues
 - Recommandations
 - Réguliers
 - Avec les clients
 - Propriétés validées
 - Vérifiabilité
 - Compréhensibilité
 - Traçabilité – d'où vient le besoin
 - Adaptabilité – peut-on changer le besoin sans changer beaucoup d'autres choses?
- Prototype
- Générer des cas de test

LYBSYS points de vue



Exemple LYBSYS – Scénari -

Initial assumption: The user has logged on to the LIBSYS system and has located the journal containing the copy of the article.

Normal: The user selects the article to be copied. He or she is then prompted by the system to either provide subscriber information for the journal or to indicate how they will pay for the article. Alternative payment methods are by credit card or by quoting an organisational account number.

The user is then asked to fill in a copyright form that maintains details of the transaction and they then submit this to the LIBSYS system.

The copyright form is checked and, if OK, the PDF version of the article is downloaded to the LIBSYS working area on the user's computer and the user is informed that it is available. The user is asked to select a printer and a copy of the article is printed. If the article has been flagged as 'print-only' it is deleted from the user's system once the user has confirmed that printing is complete.

Exemple LYBSYS - Scénari -

What can go wrong: The user may fail to fill in the copyright form correctly. In this case, the form should be re-presented to the user for correction. If the resubmitted form is still incorrect then the user's request for the article is rejected.

The payment may be rejected by the system. The user's request for the article is rejected.

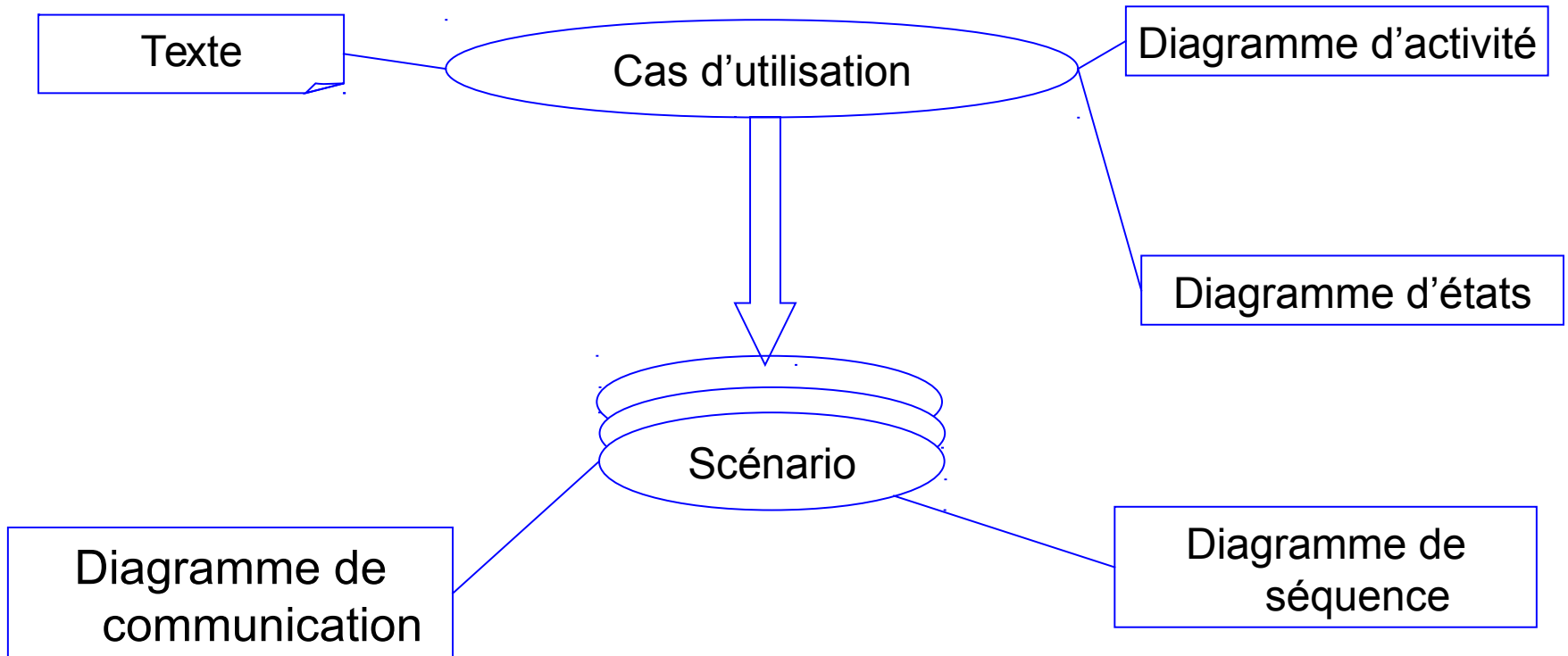
The article download may fail. Retry until successful or the user terminates the session.

It may not be possible to print the article. If the article is not flagged as 'print-only' then it is held in the LIBSYS workspace. Otherwise, the article is deleted and the user's account credited with the cost of the article.

Other activities: Simultaneous downloads of other articles.

System state on completion: User is logged on. The downloaded article has been deleted from LIBSYS workspace if it has been flagged as print-only.

Diagrammes



Cas d'utilisation

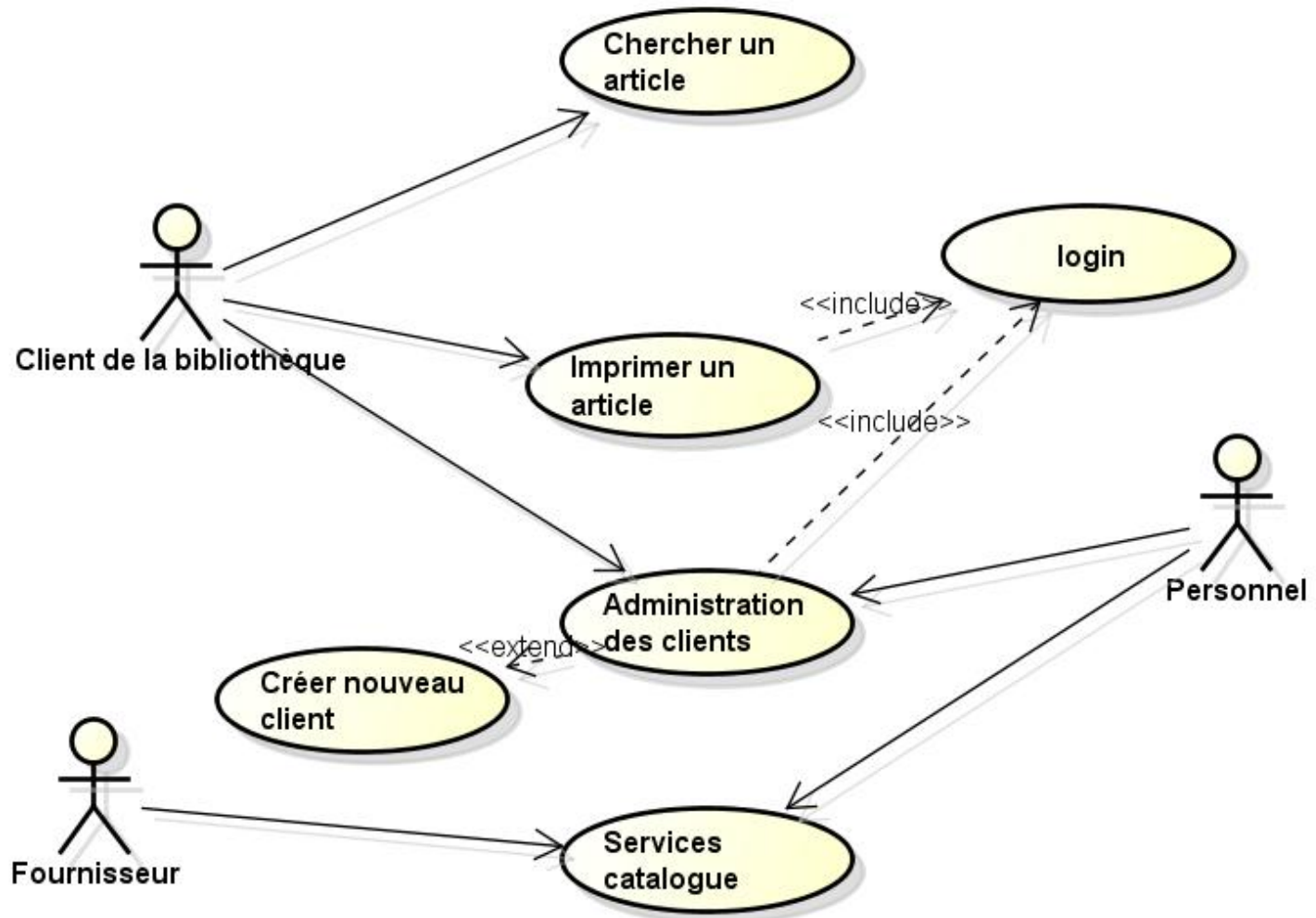


Diagramme de séquence

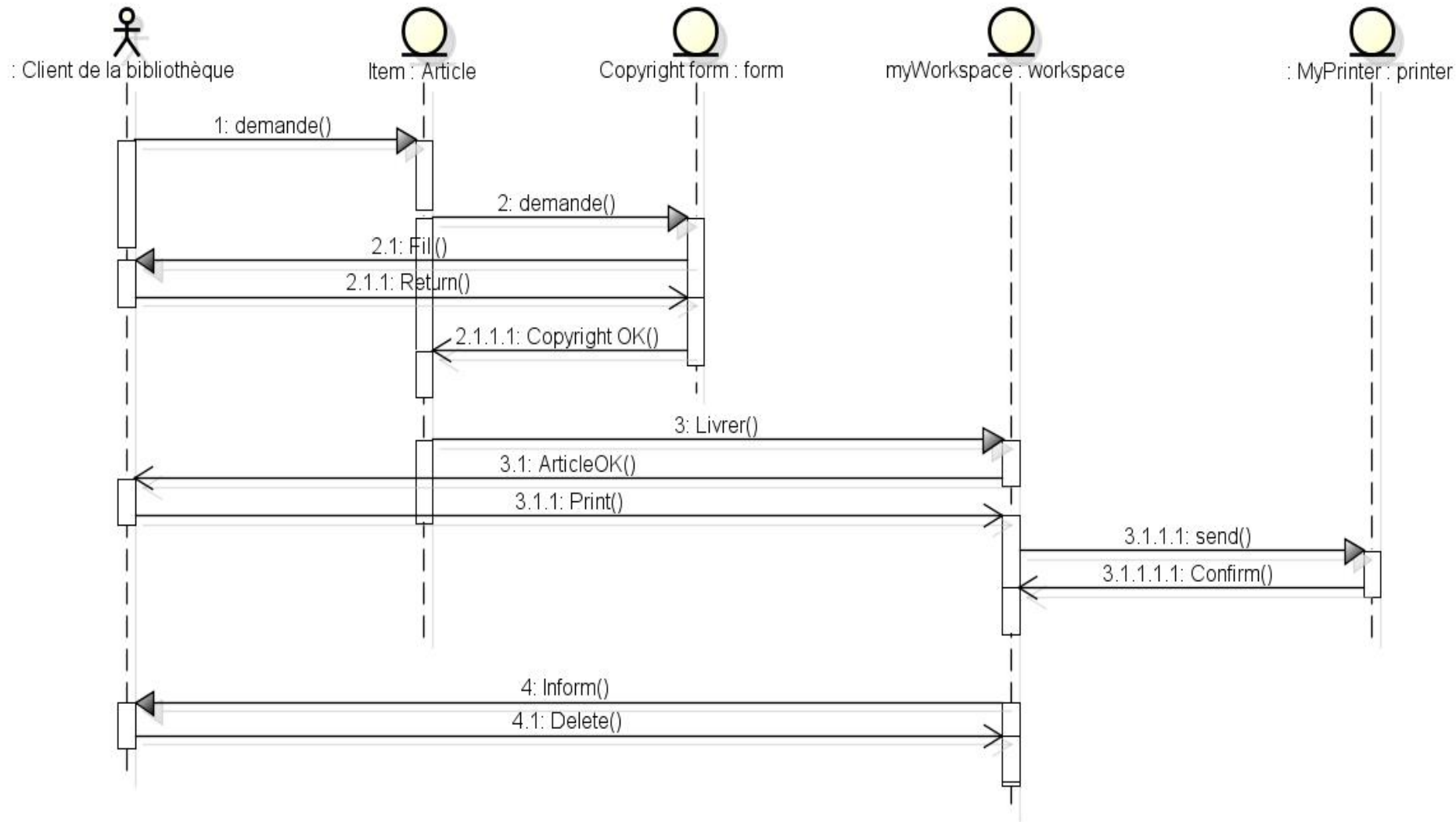


Diagramme de communication

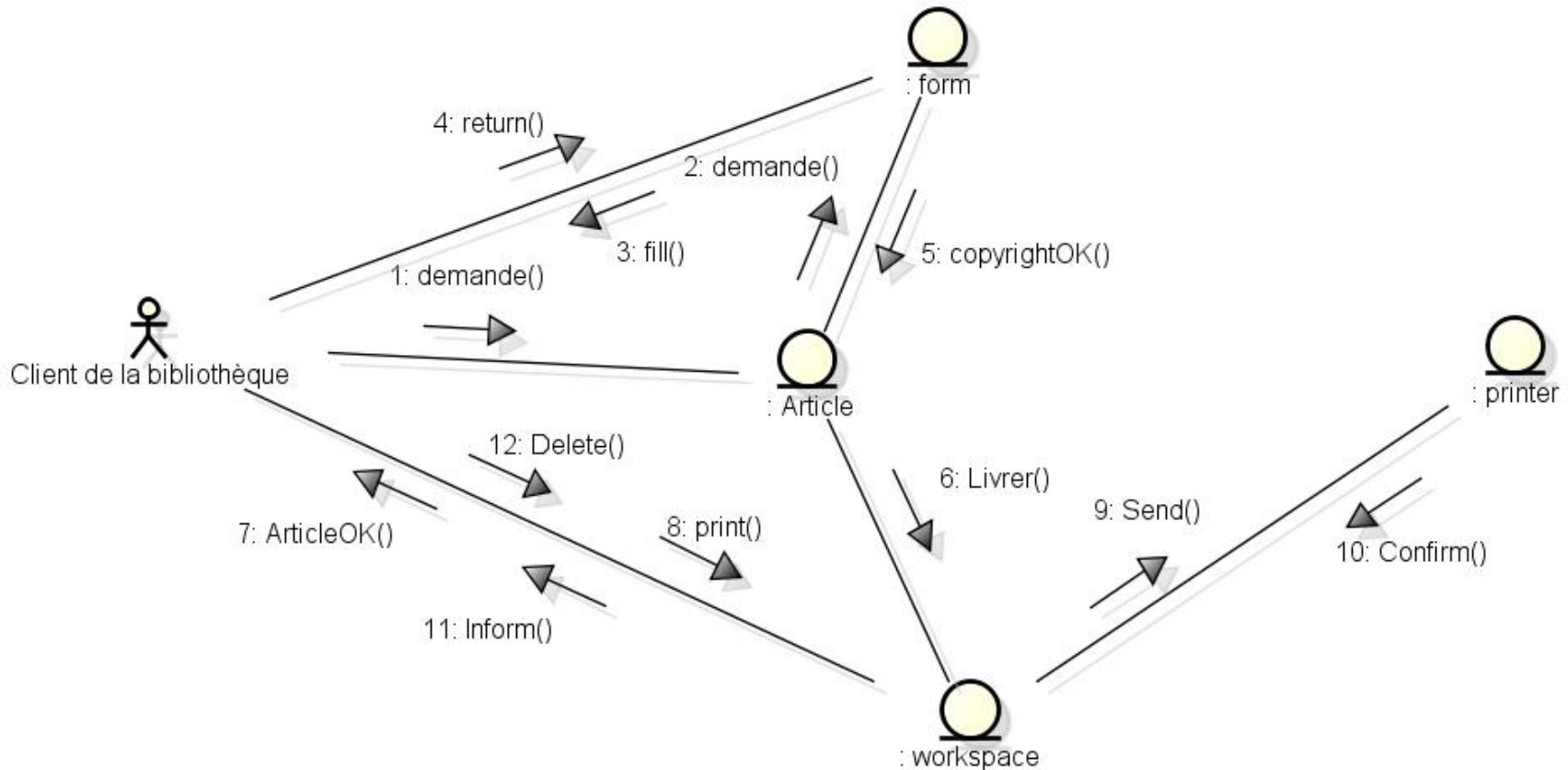


Diagramme d'activité

