

# RAPPORT DE TP UML

Professeur : A. Vailly

Dossier et TP réalisés par

Zone rendue volontairement anonyme par souci de confidentialité

---

## SOMMAIRE

Introduction .....	3
1. Diagrammes de Cas d'utilisation .....	4
2. Diagrammes de Séquences et Diagrammes de Collaboration.....	10
3. Diagrammes de Classes.....	14
4. Diagrammes d'Etats Transitions.....	16
Conclusion.....	19

## Introduction

Ce projet d'ULM a pour but de nous initier aux principales méthodes et diagrammes utilisés par la méthode UML. Cette méthode, aujourd'hui très populaire, est beaucoup utilisée dans de nombreux domaines.

Nous présentons, dans ce dossier, avec comme support le sujet "PHILDEX", quelques une des méthodes possibles en UML. Ces méthodes sont très diverses dans leur formalisme comme dans leur intérêt vis-à-vis de la conception ou de l'analyse.

Nous essaierons d'organiser ce rapport de façon logique en proposant les méthodes en fonction de leur niveau d'analyse (Elles seront donc présentées par complexité croissante). Après chaque méthode (type de schéma) nous essaierons de formuler une critique sur l'intérêt de celle-ci, en prenant notamment en compte les deux critères suivants :

- En premier lieu, le caractère "compréhensible" du schéma pour une personne non initiée. (Très utile pour toute relation entre informaticiens (fournisseurs de services) et non informaticiens (donneurs d'ordres).
- Ensuite, la complétude du schéma, et donc son intérêt pour son analyse et la conception.

Il sera ensuite proposée une conclusion sur l'apport de la méthode UML sur le travail d'analyse et conception, ainsi qu'une conclusion sur l'apport de ce TP sur le plan personnel.

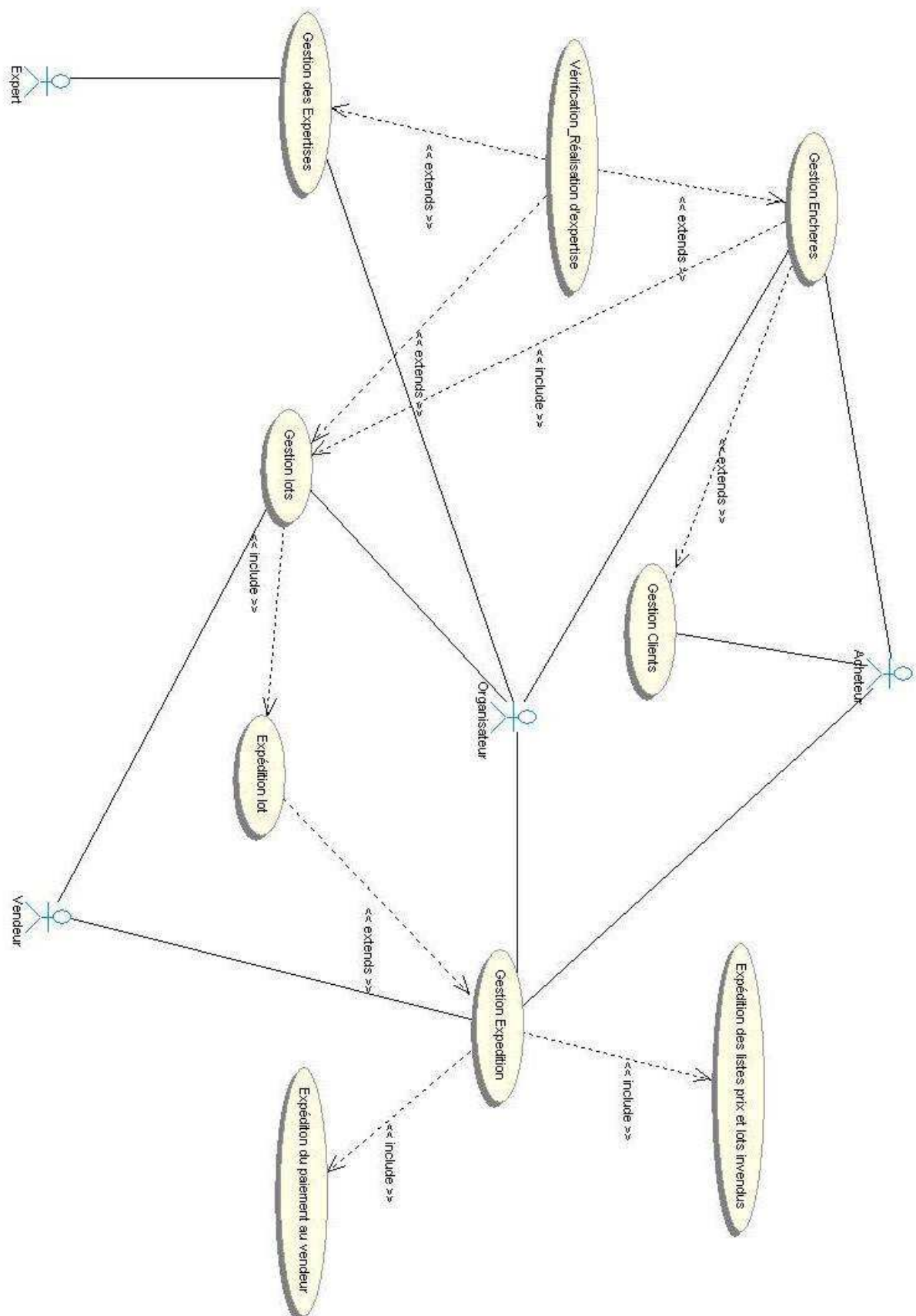
## 1. Diagrammes de Cas d'utilisation

Le premier diagramme que nous présentons est celui des cas d'utilisation. Ce diagramme est en effet, celui qui est, à notre avis, le plus simple dans son formalisme, facilement compréhensible pour un non informaticien. Il n'est pas pourtant dénué d'intérêt pour la conception. Il forme selon nous un très bon outil pour une première analyse et formalisation des besoins exprimés par les donneurs d'ordre. Le diagramme est présenté en page suivante.

La conception étant un travail très itératif, nous présentons, à titre d'exemple, en annexes, deux étapes de raffinement du cas d'utilisation présenté ci avant.

La caractéristique principale de ce schéma est donc de partir de situations pratiques d'utilisation de la future l'application, ou de règles de gestion, pour découvrir les tâches qui seront nécessaires dans l'application. Il est donc très "ludique" et permet au donneur d'ordre de participer activement à l'élaboration de la conception.

Sa principale limite découle de sa principale qualité, à savoir sa simplicité, à la fois graphique et conceptuelle, qui ne permet pas d'aller suffisamment loin dans la conception. Il doit donc être complété. Les descriptions textuelles associées aux cas d'utilisation sont une réponse à la relative superficialité de ce formalisme. Nous avons choisi de présenter trois descriptions textuelles, correspondant à trois situations des cas d'utilisation. Pour des raisons de mise en page, ces descriptions se trouvent deux pages ci après.



Description textuelle du cas d'utilisation "Gestion clients"

<b>Cas d'utilisation gestion clients.</b>	
Acteurs primaires : Acheteurs, Vendeurs. Acteurs secondaires : Organisateur	
Invariant : Un client est inscrit une seule fois.	
<p style="text-align: center;">Description</p> <p>Un client est une personne physique étant intéressée (potentiellement) par les lots mis en vente. Elle reçoit à ce titre un catalogue avant le début des enchères. La gestion de la base de données des clients consiste à enregistrer les nouveaux clients, à en modifier la description le cas échéant, et à supprimer un client ne souhaitant plus l'être. Chaque client obtient un numéro de client unique.</p> <p><u>Cas inscription :</u> Il s'agit d'inscrire dans la base les coordonnées du nouveau client. Il lui est attribué un numéro de client automatiquement.</p> <p><u>Cas Mise à Jour :</u> Il s'agit de modifier les informations relatives à un client pour les mettre à jour.</p> <p><u>Cas suppression :</u> Il s'agit de supprimer un client qui ne souhaite plus figurer parmi les clients et recevoir le catalogue.</p>	
<p style="text-align: center;">Exceptions</p> <p><u>Cas inscription :</u> Précondition : Client radié Résultat : Interdire l'inscription</p> <p><u>Cas Mise à Jour :</u> Précondition : Données erronées. Résultat : Non réalisation de la mise à jour.</p> <p><u>Cas suppression :</u> Précondition : Client inconnu. Résultat : Non suppression du client inexistant. Précondition : Présence d'impayés. Résultat : Interdiction de la suppression tant que le client n'a pas tout réglé.</p>	

Description textuelle du cas d'utilisation "Gestion des enchères"

<b>Cas d'utilisation : gestion des enchères</b>	
Acteurs primaires : Acheteurs Acteurs secondaires : Organisateur	
Invariant : Une enchère est faite sur un lot précis Un vendeur ne peut enchérir sur ses lots.	
<p style="text-align: center;">Description</p> <p>Pendant la durée d'ouverture des enchères, une personne inscrite comme client peut, déposer une enchère sur un lot. Une enchère comprend un prix et optionnellement une demande d'expertise. Le prix marqué sur l'enchère gagnante sera retenu pour le calcul du prix final du lot. Après écoulement du délai, les enchères sont clôturées, et un gagnant est désigné pour chaque lot ayant fait l'objet d'une enchère valide.</p> <p><u>Cas enchère correcte :</u> Précondition : enchère correctement réalisée Résultat : Enregistrement de l'enchère.</p> <p><u>Cas enchère réalisée après la clôture :</u> Précondition : enchère réalisée après clôture Résultat : Interdire l'enchère.</p> <p><u>Cas enchères de prix égal à la clôture :</u> Précondition : deux enchères « gagnantes » à égalité après clôture. Résultat : Tirage au sort du gagnant.</p>	
<p style="text-align: center;">Exceptions</p> <p><u>Cas enchère réalisée par le vendeur du lot :</u> Précondition : Enchérisseur = Vendeur Résultat : Interdire l'enchère.</p> <p><u>Cas enchère réalisée par personne non cliente :</u> Précondition : Enchérisseur non enregistré comme client. Résultat : Enregistrement du client</p>	

Description textuelle du cas d'utilisation "Expéditions"

<b>Cas d'utilisation Expéditions.</b>	
Acteurs primaires : Organisateur Acteurs secondaires : Acheteurs, Vendeurs	
Invariant : Après clôture des enchères, - Expédition aux nouveaux clients des lots payés. - Expédition aux anciens clients des lots gagnés par eux.	
<p style="text-align: center;">Description</p> <p>La gestion des expéditions comprend l'envoi des lots et factures aux clients de la maison Phildex, après que les éventuelles expertises sur les lots soient rendues, ainsi que l'envoi du paiement aux vendeurs. La gestion des expéditions a aussi en charge l'expédition des listes des prix de vente et la liste des lots invendus.</p> <p><u>Cas envois des lots aux Nouveaux Clients :</u> Il s'agit d'envoyer les lots aux nouveaux clients qui ont retourné le paiement correspondant au montant de leur facture.</p> <p><u>Cas envois des lots aux Anciens Clients :</u> Il s'agit d'envoyer aux Anciens clients leurs lots, indépendamment de la réception des paiements.</p> <p><u>Cas envois des paiements aux Vendeurs :</u> Suite à la réception des paiements des acheteurs, le paiement aux vendeurs consiste à donner aux vendeurs la somme correspondant au montant total de leurs lots vendus.</p> <p><u>Cas envois courants :</u> Il s'agit de l'envoi de l'ensemble des factures aux clients. Sont aussi envoyés, suite à la clôture des enchères, une liste des lots invendus, et une liste des prix de vente.</p>	
<p style="text-align: center;">Exceptions</p> <p><u>Cas retour du courrier (NPAI) :</u> Précondition : Retour d'un courrier, motif NPAI Résultat : Contacter le client et MAJ de sa « fiche ».</p> <p><u>Cas grève Service Postal :</u> Précondition : Préavis de grève déposé au service postal. Résultat : Retarder l'envoi jusqu'à rétablissement de la situation.</p> <p><u>Cas lot perdu par le Service Postal :</u> Précondition : Lot perdu par le Service Postal. Résultat : Demande de remboursement du service postal. Remboursement du client après accord avec lui.</p>	



Cas courrier perdu par le Service Postal :

Précondition : lettre perdue par le Service Postal.

Résultat : Ré-envoi du courrier

Cas mauvais payeur :

Précondition : Le client n'a pas payé son lot après un délai fixé

Résultat : Envoi d'une lettre de relance.

Marquage « Mauvais Payeur »

## 2. Diagrammes de Séquences et Diagrammes de Collaboration

Nous choisissons de présenter les diagrammes de séquence en deuxième méthode, car nous pensons que ceux-ci complètent les diagrammes de cas d'utilisation, en apportant un "premier jet" au concept d'interfaces/classes/objets, nécessaires à la conception, mais échappant au non informaticien.

Nous avons développé deux diagrammes de séquences, pour illustrer deux cas d'utilisations découverts avec les diagrammes du même nom.

### Premier Diagramme :

Cas d'un nouveau vendeur apportant un lot. Si le lot est accepté, le vendeur est créé et le lot enregistré. Dans le cas contraire on informe le vendeur du refus.

#### *Commentaire sur le diagramme.*

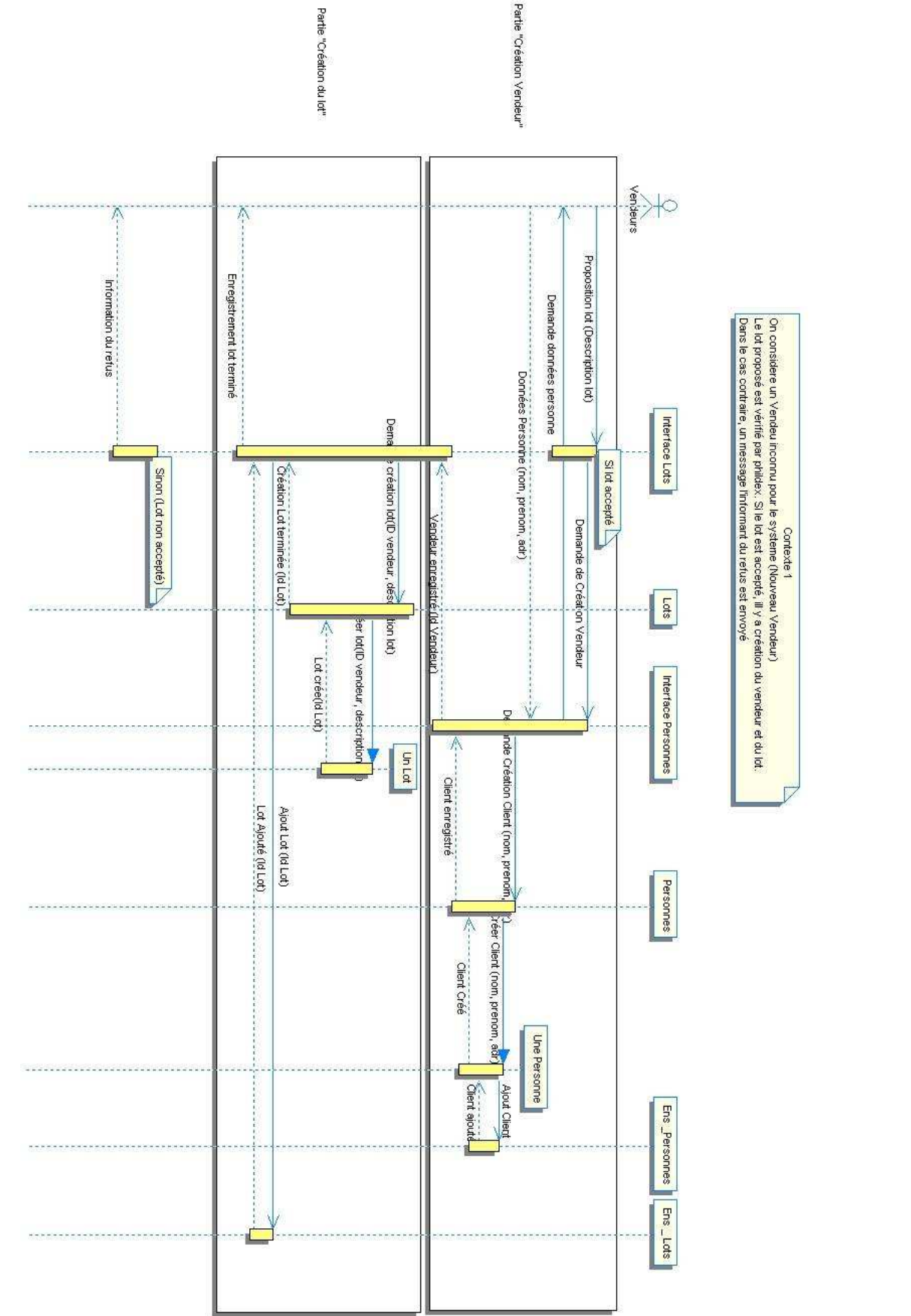
Nous avons choisi d'ajouter immédiatement un objet "personne" à son ensemble, par appel à la fonction d'ajout directement depuis l'objet créé. A contrario, pour les objets "lots" nous avons choisi que se soit la classe interface qui s'en charge. Ce découpage nous permet de créer un lot mais de ne l'ajouter à l'ensemble qu'une fois toutes les vérifications nécessaires terminées, par exemple le retour d'un résultat d'expertise. Ce découpage évite de bloquer la création du lot.

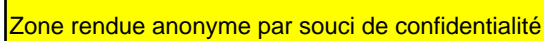
### Second Diagramme:

Cas d'un vendeur existant proposant un lot que la société Phildex désire expertiser avant mise en vente. Selon le résultat de l'expertise, le lot est enregistré ou refusé.

#### *Commentaire sur le diagramme.*

Ce diagramme a la particularité de posséder de nombreuses conditions, selon le résultat de l'exécution. De ce fait, il est assez riche, mais le diagramme est vite surchargé par ses cas imbriqués. Sa lecture n'est donc pas aussi simple que le précédent diagramme.

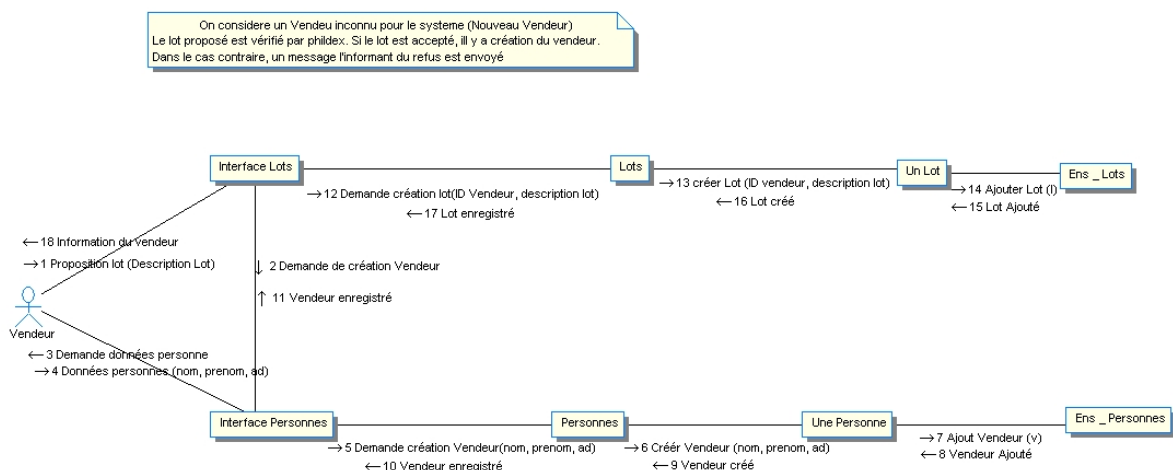




Ces diagrammes de séquence, par rapport aux Use Case, intègrent deux "nouveau" intéressantes : Tout d'abord, une approche du concept d'objet (découpage en interfaces/classes/objets), et la notion de temporalité. La grande notion d'envoi de message sur laquelle repose ce diagramme, est une "nouveau" qui découle de l'insertion du concept d'objet.

Du fait de l'intégration des ces notions très intéressantes du point de vue de la conception, le diagramme est le premier vraiment transposable dans une optique d'analyse. Le revers de la médaille est que le formalisme, sans s'alourdir trop du point de vue graphique, implique des connaissances informatique supérieures, notamment le concept Objets-Messages, et devient de ce fait plus dur à comprendre pour les non initiés.

Nous avons choisi délibérément de présenter les diagrammes de Collaboration avec les diagrammes de séquence car ces deux types de diagrammes sont très proches et intimement liées. En effet, les diagrammes de collaboration "ne sont que" des diagrammes de séquences condensés dans lesquels est en quelque sorte cachée la temporalité.



### Diagramme de collaboration modélisant la même situation que le diagramme de séquence numéro 1

Ces diagrammes de collaboration ont pour avantage de rendre présenter un contenu proche de celui des diagrammes de séquence, mais de manière plus synthétique, plus condensée. Ils peuvent donc servir de "pont" entre les diagrammes de cas d'utilisation, vraiment très simples, et des formalismes plus complets.

L'intérêt pour la conception de ces diagrammes est un peu limité par rapport à leurs équivalents diagrammes de séquence, du fait de la perte partielle de la temporalité dans le diagramme (on garde l'ordre des messages mais, le parallélisme n'est plus clairement affiché par le formalisme).

Si nous devons choisir entre l'un des deux formalismes, nous choisirions les diagrammes de séquence. En effet, le concept informatique qui peut poser problème à une personne non initiée n'est pas la temporalité, que les diagrammes de Collaboration esquivent, mais bien le concept d'objet et d'envoi de messages. Dans ce contexte, quitte à impliquer une personne non initiée dans cette démarche de conception, il nous paraît plus avantageux de le faire avec un formalisme sensiblement plus riche, et pas fondamentalement plus compliqué.

### 3. Diagrammes de Classes

Dans notre souci de présenter les méthodes par degrés de difficulté nous avons choisi de présenter en 3<sup>ème</sup> méthode le diagramme de classe. Bien que cette méthode soit basée sur des concepts difficiles pour les non initiés, nous l'avons préféré aux diagrammes d'état transition pour leur côté "statique" sans doute plus facile à assimiler. En ce qui concerne l'intérêt pour la conception, celui-ci est très grand, puisque ce diagramme est le diagramme de base du développement orienté objet.

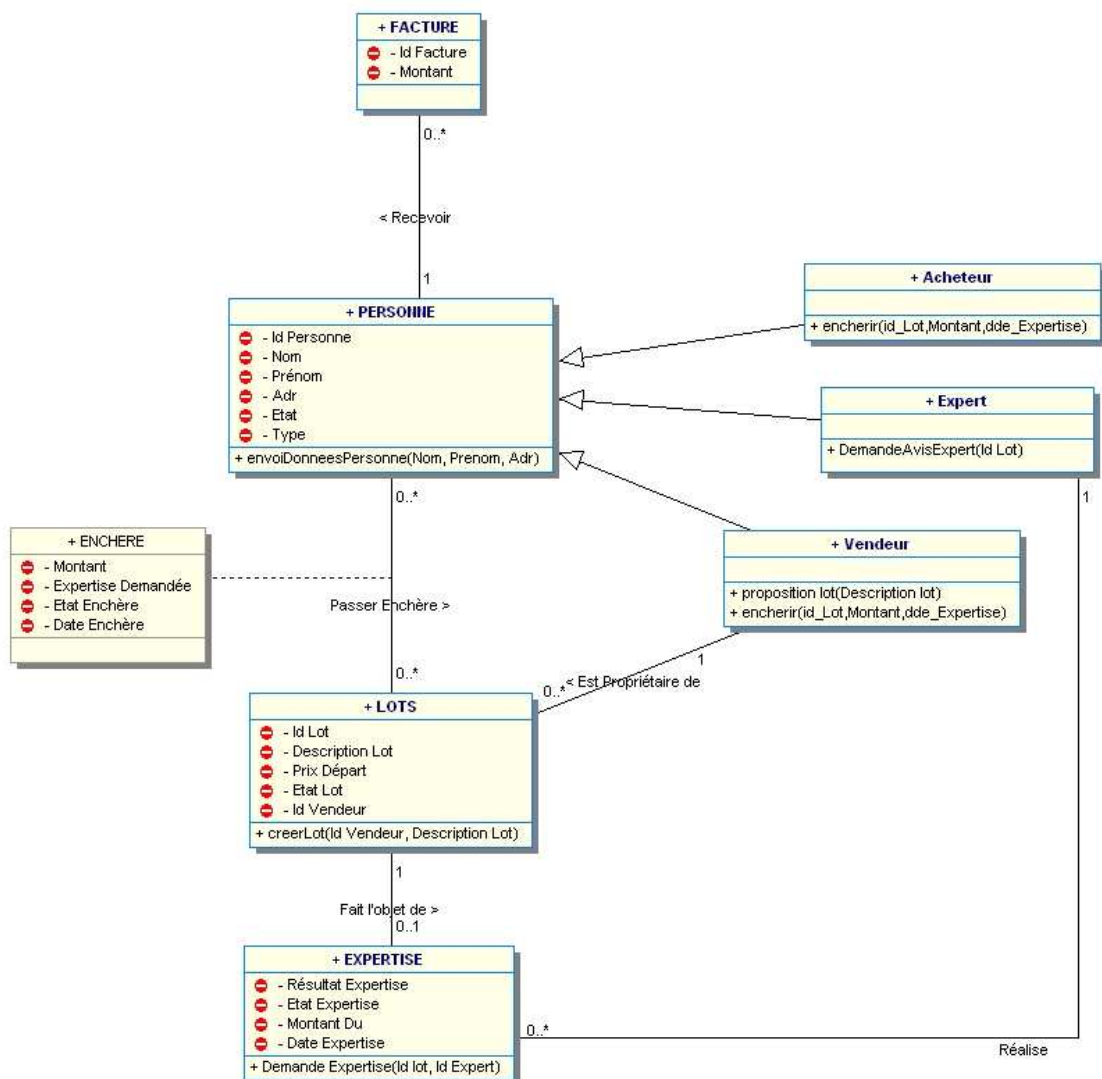


Diagramme des classes.

Certaines contraintes du système ne sont pas exprimées sur ce diagramme, notamment celles portant sur le droit d'une personne à enchérir sur un lot donné ou non. Nous avons choisi d'exprimer ces contraintes en OCL.

---

Première contrainte identifiée : Un expert ne peut pas enchérir sur un lot.

Nous nous plaçons dans le contexte Expert.

*Self.Enchere->IsEmpty()*

Cette contrainte indique bien que la liste des lots sur lesquels l'expert à enchéri est vide, c'est-à-dire qu'il n'a pas fait d'enchère.

Seconde contrainte identifiée : Un vendeur ne peut pas enchérir sur son propre lot.

Nous nous plaçons dans le contexte Vendeur

*Self.Enchere->ForAll(i :integer | Lot(i).Id Vendeur<>Self.Id Personne)*

Cette contrainte indique que pour chaque lot de la liste des lots sur lesquels le vendeur a effectué une enchère, le propriétaire du lot, n'est pas lui même.

D'un point de vue de la conception, il est clair que ce diagramme est important, puisqu'il jette les bases du découpage en objet du programme. Dans cette optique, ce diagramme est en parfaite harmonie avec les diagrammes de séquences présentés plus avant. Le diagramme de classes sera particulièrement réfléchi lors de la conception d'une application, car il conditionne nombre de choix techniques. Ainsi, une modification du diagramme de séquence peut entraîner une modification d u diagramme de classes et de tout diagramme reposant sur lui. Inversement, une modification du diagramme de classes (par les développeurs et non à la demande du client) peut entraîner une incohérence au niveau des diagrammes de séquences qu'il va falloir expliquer et faire revalider par le client.

Ce diagramme de classe nous paraît être le premier diagramme présenté ici qui ne peut pas facilement être présenté au public non informaticien. Pour nous, ce diagramme le diagramme "porte d'entrée" vers des diagrammes de plus en plus complexes (donc moins faciles pour les non initiés) mais de plus en plus riche pour la conception. Il forme donc une étape dans le processus de d'analyse et conception.

#### 4. Diagrammes d'Etats Transitions

Pour terminer, nous présentons les diagrammes d'Etat transition. Ces diagrammes sont les diagrammes les plus spécialisés à notre avis des diagrammes présentés jusqu'alors. En effet, ceux-ci montrent la dynamique du système, et non plus une vision statique du système.

Nous avons choisi de développer deux cas, assez différents des autres diagrammes pour ne pas laisser de côté tout un pan de la conception. Nous nous sommes donc ici intéressés aux offres réalisées par les acheteurs, ainsi que la "vie d'un timbre", de sa proposition à Phildex, à son acquisition par l'acheteur gagnant.

##### Premier Diagramme:

Ce diagramme modélise l'enregistrement d'un lot au catalogue avec les vérifications nécessaires.

##### *Commentaire sur le diagramme.*

La principale étape de ce diagramme est la recherche de l'identité du vendeur. En effet, pour être en cohérence avec le diagramme de Use case, en plus de l'identification d'un vendeur enregistré, il nous a été nécessaire d'identifier qu'un vendeur nouveau ne soit pas en réalité un vendeur banni. Ceci explique l'automate hiérarchique "Identification du vendeur"

##### Second Diagramme :

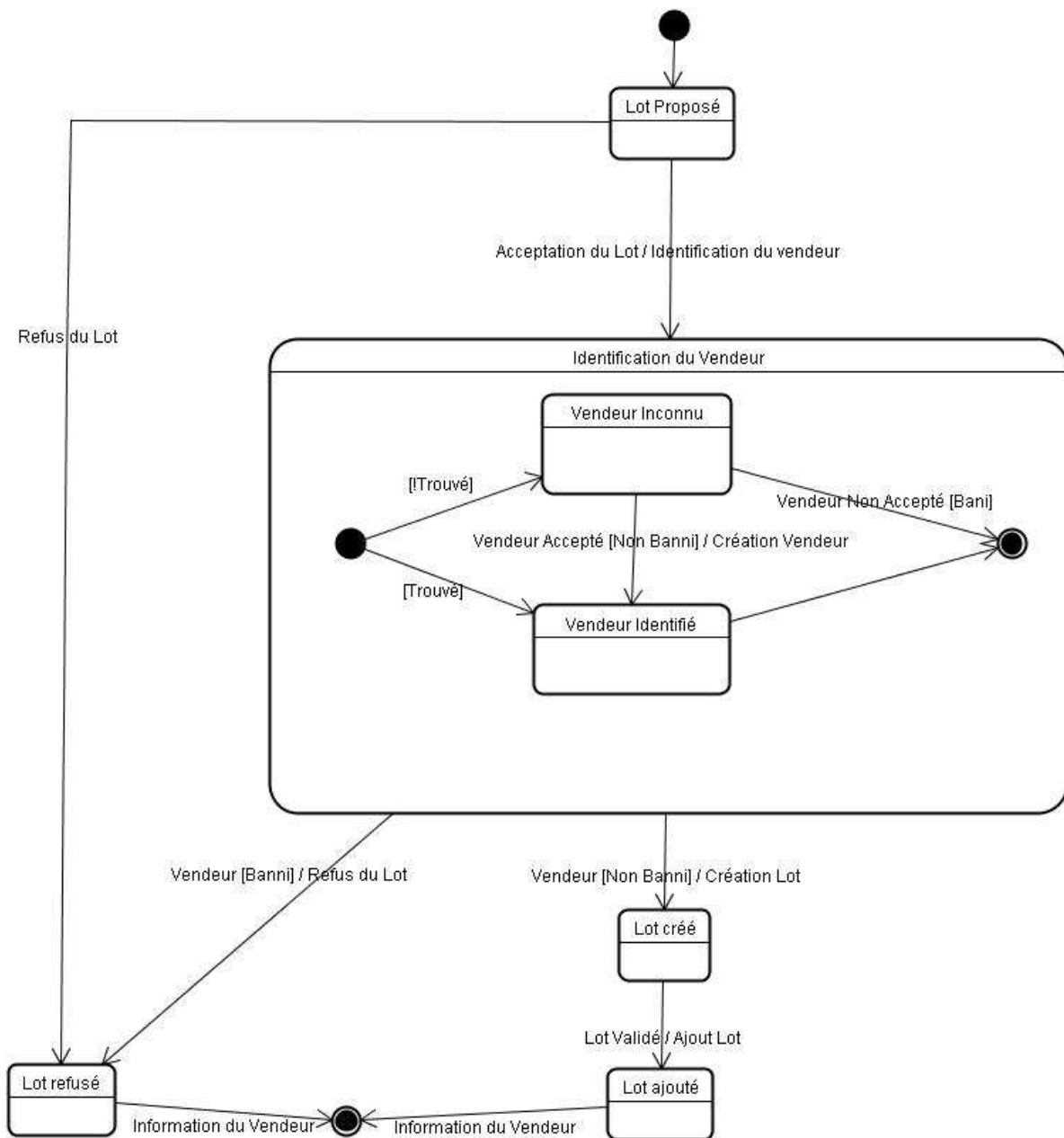
Ce second diagramme modélise les différentes étapes de la vie d'une offre effectuée par un acheteur sur un lot mis en vente.

##### *Commentaire sur le diagramme.*

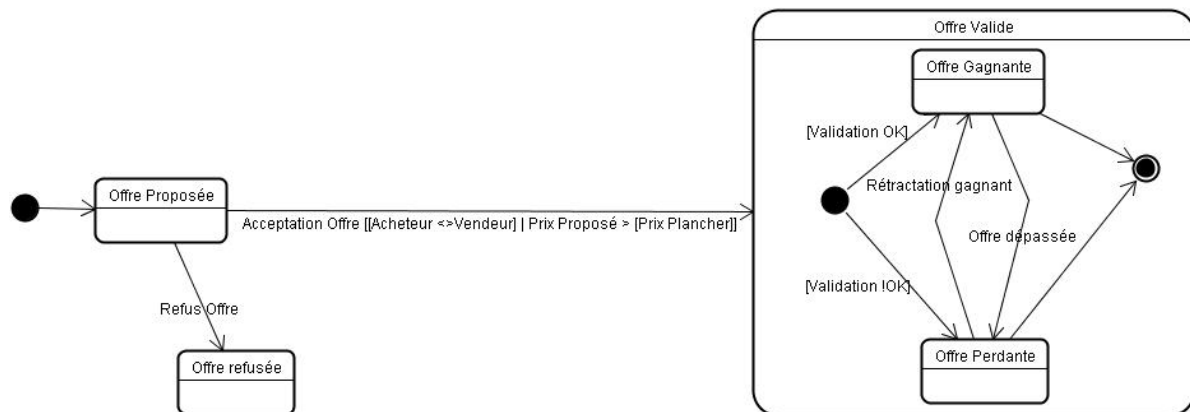
Nous aurions pu représenter ce diagramme sans faire appel aux niveaux hiérarchiques. Cependant, il nous a semblé intéressant de prendre en compte le fait qu'une offre, une fois validée, est dans un état nouveau, "Offre Valide", qu'elle soit gagnante ou perdante.

Nous avons aussi fait l'hypothèse qu'un gagnant pouvait se retirer dans le cas où l'expertise demandée par le gagnant révélait que le lot est un faux. Dans ce cas, l'enchère immédiatement perdante devient gagnante.





Automate modélisant l'enregistrement d'un lot au catalogue.



Automate modélisant la "vie d'une offre".

Les diagrammes d'états transitions sont à notre avis très intéressants du point de vue de la conception. En effet, ils présentent sous une forme dynamique la conception réalisée sous forme statique, grâce notamment aux diagrammes de classes. Ces diagrammes d'états transitions sont de puissants outils de vérification de la dynamique d'un système.

D'un point de vue de la conception donc, ces diagrammes sont très riches, car ils reposent sur le diagramme de classe, et apportent en plus l'évolution des objets représentés par ces classes. Si ces diagrammes sont insuffisants individuellement à réaliser une analyse correcte d'un système, ils peuvent aider le concepteur à repérer un problème invisible sur le diagramme des classes.

Du point de vue de la compréhension pour une personne non initiée, nous pensons que malgré le formalisme facile à appréhender, les concepts sous-jacents font du diagramme des classes un diagramme peu aisé à comprendre.

## Conclusion

Par ce TP, nous avons pu nous familiariser avec les principaux diagrammes d'UML. Nous avons vraiment pu apprécier les qualités et les faiblesses de chacun de ces diagrammes. Ce TP a été une très bonne illustration du cours.

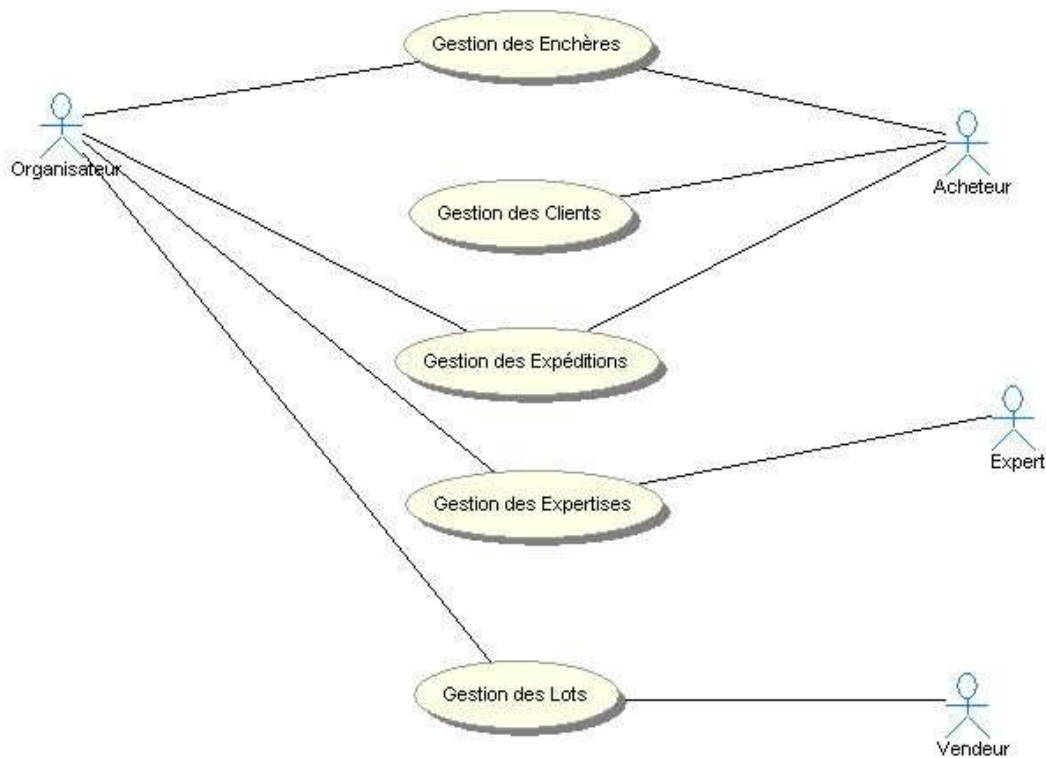
Comme toujours en conception, le temps passé à étudier le sujet et à lever des ambiguïtés, voire à poser des hypothèses lorsque celui-ci n'est pas assez précis a été important, malgré le fait que nous étions déjà familier du sujet.

Nous avons aussi pris conscience qu'il n'y a pas une conception, mais des conceptions d'un même problème. Cette multiplicité des solutions était au départ un handicap. Nous nous sommes souvent posé la question suivante: *"Avons-nous bien réfléchi à toutes les possibilités?"* ou encore *"Et si...?"*. Finalement, la multitude des réponses possibles à un même problème fait que la conception est à chaque fois un travail nouveau, ce qui en fait une pratique intéressante, puisque toujours à remettre en cause.

Nous avons choisi de classer les méthodes présentées dans un ordre d'intérêt (Cf. intro). Pour être complet, nous souhaitons apporter une nuance à notre classification. En effet, selon le type d'application développé, l'importance des méthodes présentées peut évoluer. Ainsi, pour une application temps réel, l'importance du diagramme de classe et des diagrammes d'état transition sera sans doute plus forte que celle du diagramme de cas d'utilisation (ou même de séquence).

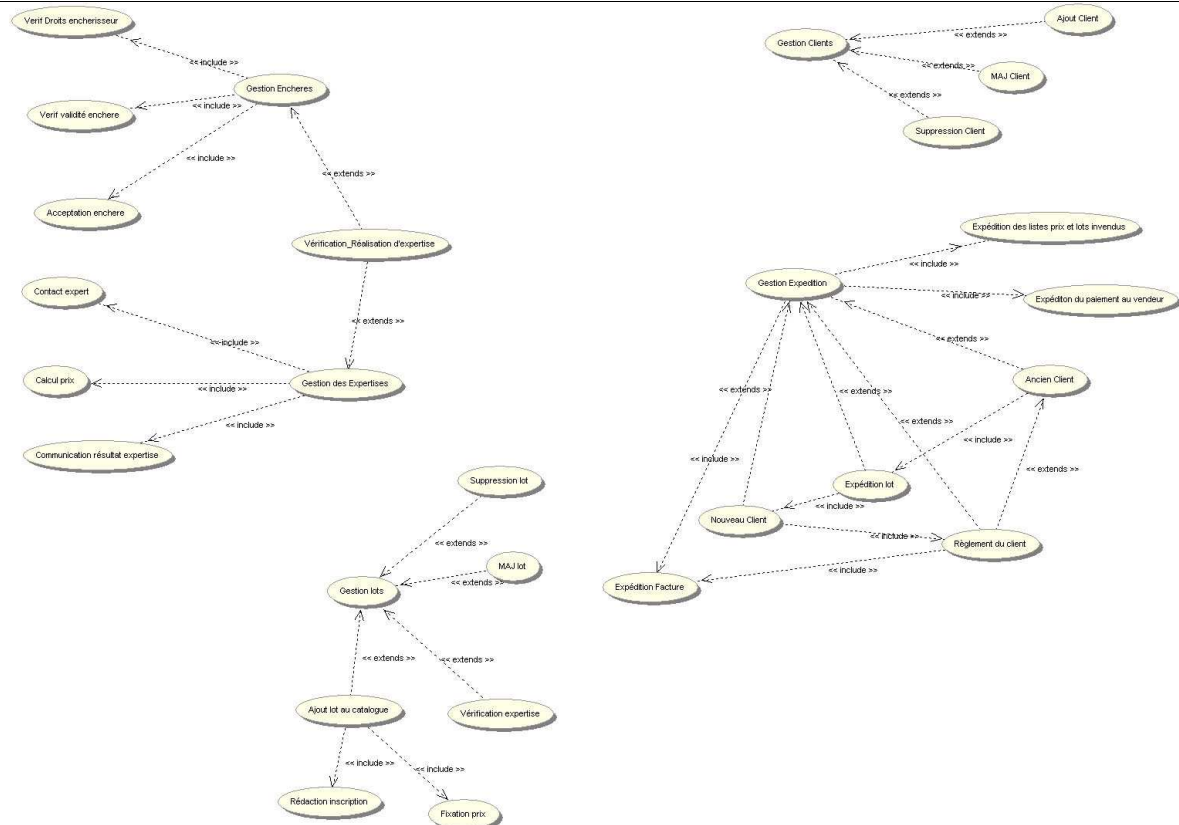
## Annexes

La conception étant un travail très itératif, nous avons jugé intéressant de montrer dans ces annexes les différentes versions qui ont abouties au Use Case présenté en section 1.



### Use Case Premiere Version.

Ce Premier Use Case est une version très haut niveau que nous avons ensuite affiné dans les versions successives.



Use Case Seconde Version.

Dans cette seconde version, nous avons trop affiné notre analyse et avons créé des Use Case trop précis. Cette Version est en outre inachevée, car nous avons ressenti rapidement que le niveau de détail développé était trop important.

La version finale, présentée en partie 1 est un "mix" entre la vision très haut niveau du diagramme 1 et la vision trop bas niveau de la version 2.