

Projet d'UML  
*Cas PHILDEX*

Zone rendue anonyme par souci de confidentialité

## Sommaire

<b><u>I.</u></b>	<b><u>Introduction</u></b>	<b>1</b>
<b><u>II.</u></b>	<b><u>Modélisation</u></b>	<b>2</b>
A.	Cas d'utilisation	2
1.	Première approche	2
2.	Cas d'utilisation avant la modélisation des diagrammes de scénarios	3
3.	Cas d'utilisation finale	5
4.	Description textuelle	5
B.	Diagrammes de classes	12
C.	Diagrammes de scénarios, séquences et collaboration	16
1.	Gestion des ventes	16
2.	Gestion des expertises	42
3.	Gestion des règlements d'achat	44
4.	Payement des vendeurs	52
5.	Mise à jour des Personnes et des Lots	52
D.	Diagramme états transition	56
1.	Lot	56
2.	Personne	57
3.	Expert	58
4.	Vendeur	58
5.	Acheteur	59
6.	Enchère	60
7.	Organisateur	61
8.	Facture	62
9.	ReglementAchat	62
10.	ReglementVente	63
E.	Problèmes rencontrés, évaluation du travail et critiques	63
<b><u>III.</u></b>	<b><u>Conclusion</u></b>	<b>66</b>

## I. Introduction

L'objectif de ce travail est de créer un système d'information permettant de la gestion automatisée d'un magasin, Phildex, spécialiste de vente aux enchères par correspondance de timbres-poste de collection.

Le logiciel doit permettre :

- La gestion des clients.
- Le stockage des données sur les lots.
- Un suivi par jour des offres.
- Le choix **automatique** d'un gagnant à la fin de période d'enchères.
- La gestion des expertises : frais, résultats.
- Le suivi des différents règlements et la gestion des échéances.

Le logiciel, en plus de ces fonctionnalités doit respecter des contraintes liées au fonctionnement courant de Phildex.

Le but de ce TP est de modéliser ce logiciel selon la méthode UML. Nous allons donc vous présenter différents diagrammes ; en expliquant pour chacun d'eux la démarche suivie et les différents choix et hypothèses que nous avons eu à faire pour obtenir le résultat final.

## II. Modélisation

### A. Cas d'utilisation

Le dégagement des différents cas d'utilisation est le résultat de notre analyse des besoins exprimés dans le sujet. En effet, cette partie reflète notre compréhension du sujet et de comment interagissent les différents acteurs qui interviennent dans le fonctionnement de l'entreprise : l'organisateur, les vendeurs, les acheteurs et les experts.

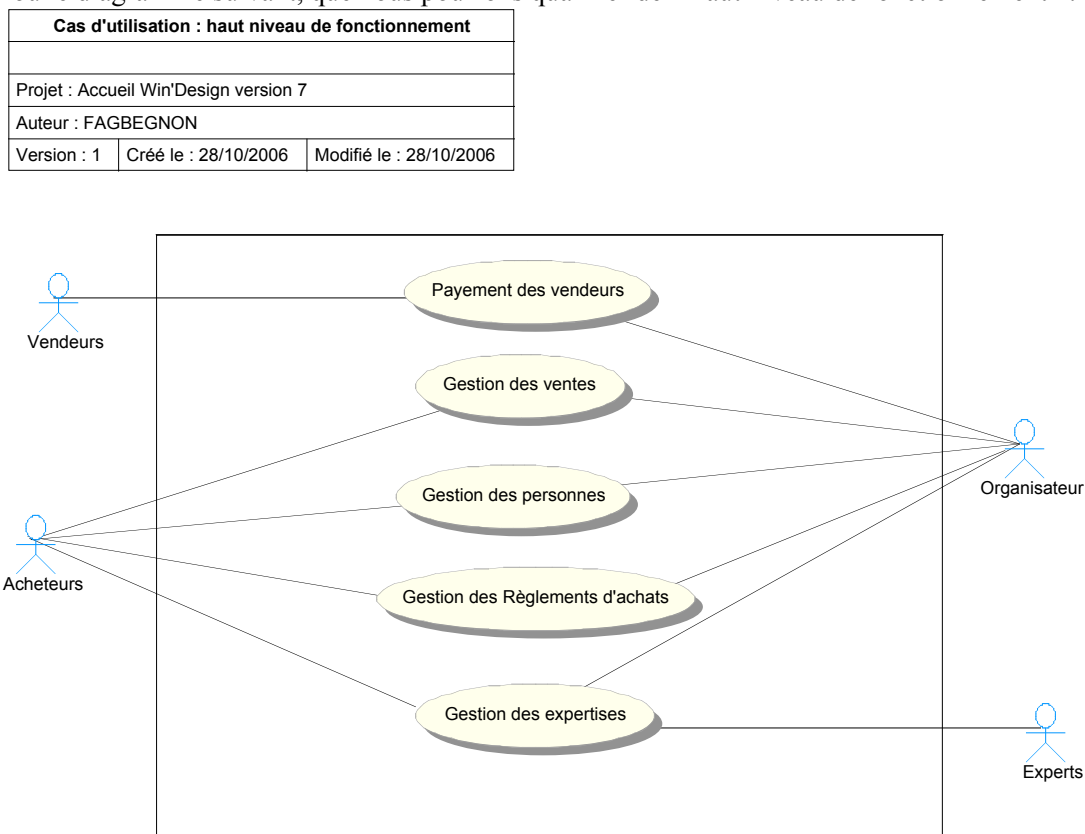
#### 1. Première approche

Notre première approche fut de savoir :

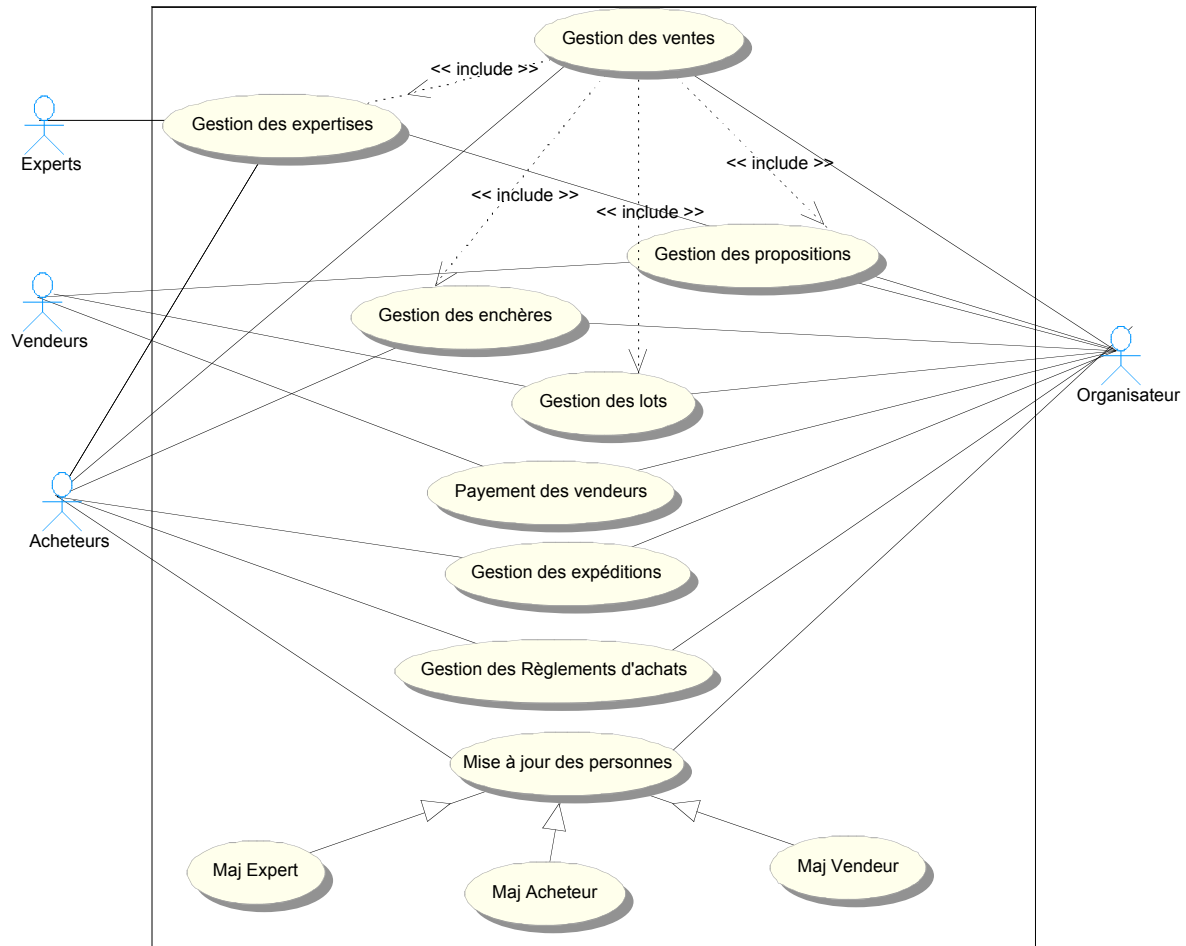
- Que doit permettre l'application ?
- Qui fait quoi et avec qui ?

Alors l'application doit principalement permettre de gérer la vente des lots, les expertises des lots, les différents règlements et gérer les données concernant certains acteurs : les acheteurs, les vendeurs et les experts.

D'où le diagramme suivant, que nous pourrions qualifier de « haut niveau de fonctionnement ».



Ce diagramme ne tient pas compte du système c'est-à-dire que les acteurs qui interviennent dans chaque cas ne sont pas forcément les utilisateurs finaux du système. Pour le cas « gestion des expertises » par exemple, nous avons considéré ici que l'acheteur est l'acteur principal car c'est lui qui fait la demande de l'expertise et que l'expertise proprement dite ainsi que sa gestion sont effectuées par l'organisateur et le ou les experts (s). Mais on verra que ce n'est pas l'acheteur l'acteur principal du point de vue « utilisation du système ».

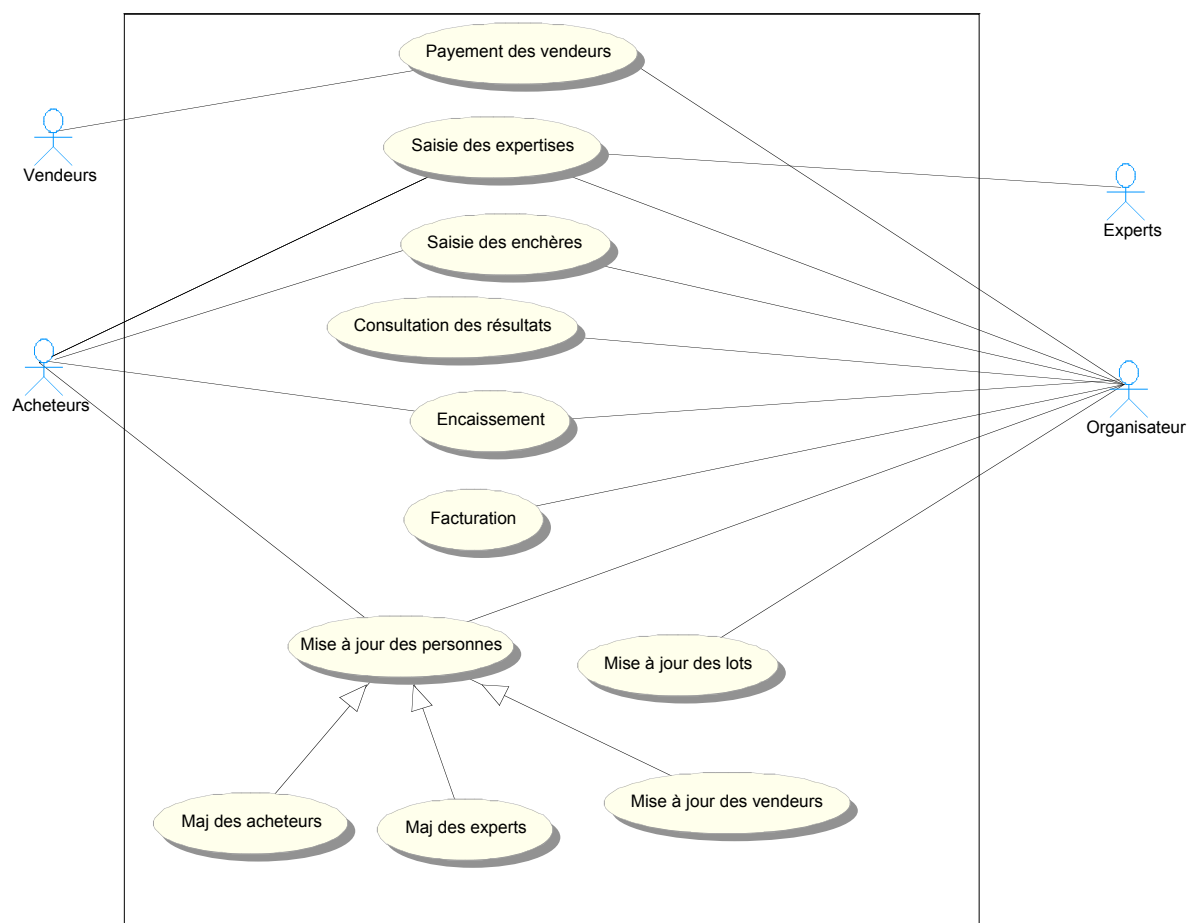


Pour la gestion des expéditions qui consiste pour nous à la facturation et à l'envoi des lots ; le système ne sera utilisé que pour la facturation, l'envoi ne demandant pas de traitement particulier si ce n'est la consultation des acheteurs gagnants, pour récupérer leurs coordonnées, et des résultats.

Décisions :

- Renommer des cas dont les noms ne sont pas assez précis.
- Supprimer les cas qui ne rentrent pas dans le fonctionnement du système.
- Prévoir un cas de consultation des résultats.

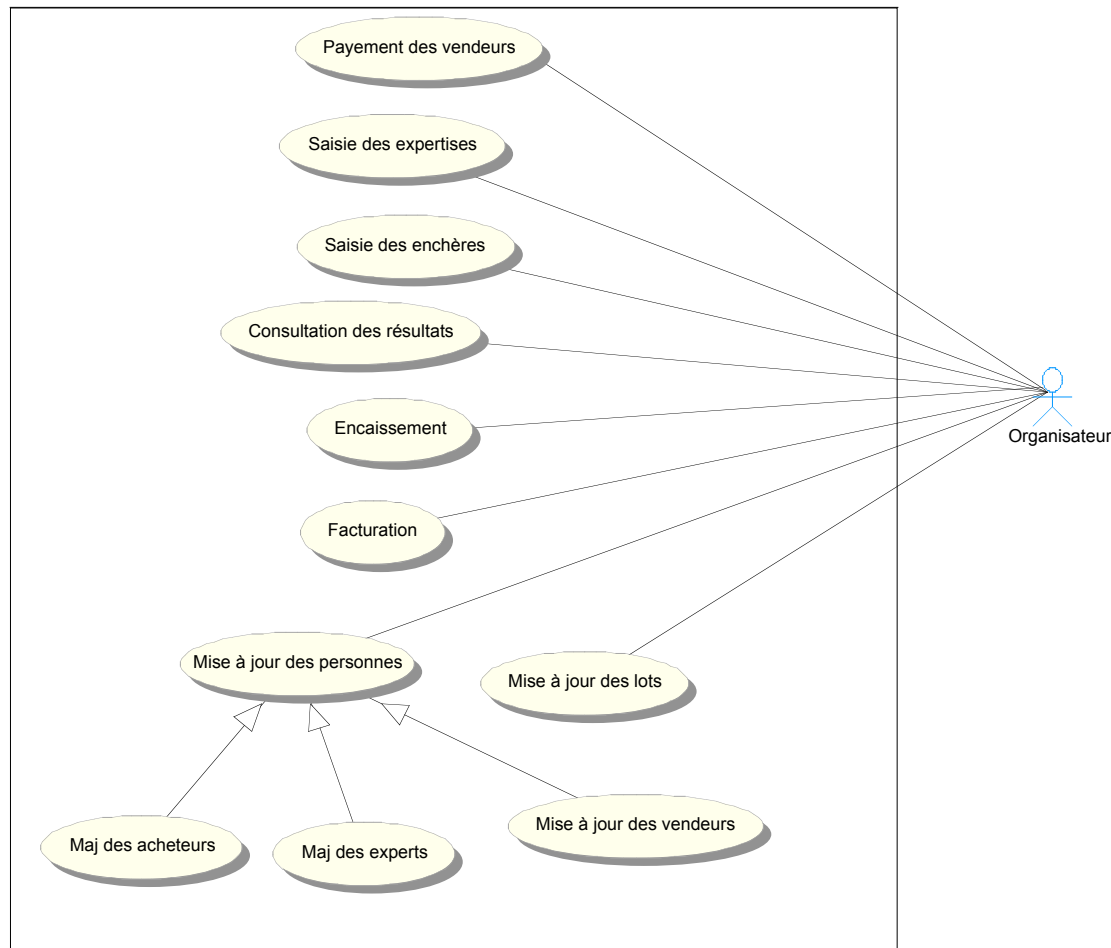
Nous avons également décidé de supprimer le cas gestion des ventes. En effet, lors de la description textuelle, nous avons remarqué que c'est un cas qui ne fait rien en particulier mais qui se décompose à la gestion des expertises, des enchères.... Ce n'est donc pas une dépendance. Nous avons pensé à la « spécialisation » mais nous avons trouvé que c'était inapproprié dans le sens où la gestion des expertises par exemple n'est pas un cas particulier de gestion des ventes.



Remarque : Le cas « gestion des Règlements d'achats », pour être plus précis, est renommé Encaissement.

### 3. Cas d'utilisation finale

La réalisation de quelques scénarios nous a fait comprendre que seul l'organisateur est « agit » sur le système. Il est le seul utilisateur, même si la source de ses utilisations peut être les experts, les acheteurs ou les vendeurs. Or les acteurs des cas d'utilisation sont ceux qui vont s'en servir. Nous avons donc supprimé quelques acteurs pour n'en laisser qu'un, l'organisateur.



Remarque : Si le système tournait sur un serveur, auquel accèdent les acheteurs pour saisir eux même leurs offres, des experts pour saisir les expertises ou des vendeurs pour proposer des lots, il aurait bien sûr fallu les garder et les reliés aux différents cas dont ils sont acteurs.

### 4. Description textuelle

#### a) Saisie des enchères (offres)

C'est l'enregistrement des offres. C'est une étape très importante dans la gestion courante de Phildex.

Pour garder la cohérence dans les données, le système doit effectuer des contrôles : vérifier si la personne est connu sinon on la crée en tant qu'acheteur, vérifier si le lot est connu sinon on annule l'opération.

Des contraintes liées aux règles de gestion de la maison doivent aussi être respectées : le prix de l'offre doit être supérieur au prix plancher, si l'acheteur a déjà enchéri sur le lot le nouveau prix proposé doit être supérieur à ceux proposés avant et le lot doit être en vente. Un vateur ne peut enchérir sur son propre lot.

<b>UC Saisie des enchères</b>
Spécialise : Gestion des ventes Include : Extend :
Invariant : le client est connu dans la base et autorisé à enchérir sur le lot, Lot connu et doit être en vente. Enchère valide : prix inférieur au prix précédemment proposé s'il y a d'autres enchères ou prix inférieur au prix plancher.
Acteur principal : Organisateur Acteur secondaire :
<b>Description</b> La gestion des enchères comprend à saisir les offres des clients
<b>Exceptions</b> <ul style="list-style-type: none"> <li>- le client inconnu dans la base : Enregistrer le client puis l'enchère</li> <li>- le client n'est pas autorisé à enchérir car il est vendeur et enchérit sur un de ses lots : Enchère refusée</li> <li>- Lot inconnu : refuser l'enchère</li> <li>- Lot non en vente : refuser l'enchère</li> <li>- Enchère invalide : refuser l'enchère</li> </ul>

## b) Clôture des ventes

<b>UC Clôture des ventes</b>
Spécialise : Include Extend
Invariant : il doit avoir au moins une enchère sur les lots mis en ventes
Acteur principal : Organisateur Acteur secondaire :
<b>Description</b> La clôture des ventes consiste à désigner un gagnant et à lancer les expertises demandées.
<b>Exception</b> Pas d'enchère sur un lot : le lot est remis en vente. Son état (en vente) reste donc inchangé.



### c) Saisie resultat des expertises

Cette fonctionnalité permet de consulter des listes suite à la clôture d'une vente. Elle n'a donc pas d'invariant

<b>UC Saisie des résultats des expertises</b>
Spécialise : Include : Extend :
Invariant : - L'expert est connu par le système et lot connu par le système. - L'expertise existe.
Acteur principal : Organisateur
<b>Description</b> La saisie des expertises consiste à entrer pour une expertise donnée les résultats reçus de l'expert : -
<b>Exceptions</b> - Expert inconnu : annuler saisie - Lot inconnu : annuler saisie - L'expertise n'existe pas: annuler la saisie
<b>UC Consultation des résultats</b>
Spécialise : - Include - Extend
Invariant : Il faut qu'il y ait au moins une enchère à traiter
Acteur principal : Organisateur
<b>Description</b> La consultation des résultats consiste à la génération et l'affichage des listes des gagnants, des lots gagnés, des lots invendus, des lots gagnés à expertiser ...
<b>Exception</b> - Si pas d'enchère : affichage de liste(s) vide(s)

### d) Consultation des expertises

La consultation des expertises a les mêmes invariants que la saisies des résultats :

- L'expert est connu et lot sont connus par le système.
- L'expertise existe.

En cas d'erreur on annule la consultation.

**e) Facturation des achats**

<b>UC Facturation des achats</b>
Spécialise :
Invariant : <ul style="list-style-type: none"> <li>- L'acheteur doit être connu.</li> <li>- Il doit avoir enchérit au moins une fois lors de la vente.</li> <li>- Il doit avoir au moins une enchère gagnante.</li> </ul>
Acteur principal : Organisateur
<b>Description</b> La facturation des achats consiste à calculer les montants à payer par un acheteur après la clôture des ventes et à les stocker.
<b>Exception</b> <ul style="list-style-type: none"> <li>- L'acheteur doit être connu : affichage d'un message d'erreur puis arrêt du traitement.</li> <li>- Il doit avoir enchérit au moins une fois lors de la vente : affichage d'un message correspondant puis arrêt du traitement.</li> <li>- Il doit avoir au moins une enchère gagnante : affichage d'un message puis arrêt du traitement.</li> </ul>

**f) Encaissement des règlements d'achats**

<b>UC Encaissement des Règlements d'achats</b>
Spécialise : <ul style="list-style-type: none"> <li>- Include</li> <li>- Extend</li> </ul>
Invariant : <ul style="list-style-type: none"> <li>- La facture correspondant au règlement doit exister.</li> <li>- L'acheteur doit être connu.</li> <li>- Il doit avoir enchérit au moins une fois lors de la vente.</li> <li>- Il doit avoir au moins une enchère gagnante.</li> </ul>
Acteur principal : Organisateur
<b>Description</b> Le règlement des achats consiste à enregistrer les règlements d'achats des lots gagnés.
<b>Exception</b> <ul style="list-style-type: none"> <li>- La facture correspondant au règlement doit exister : affichage d'un message d'erreur puis arrêt du traitement.</li> <li>- L'acheteur doit être connu : affichage d'un message d'erreur puis arrêt du traitement.</li> <li>- Il doit avoir enchérit au moins une fois lors de la vente : affichage d'un message correspondant puis arrêt du traitement.</li> <li>- Il doit avoir au moins une enchère gagnante : affichage d'un message puis arrêt du traitement.</li> </ul>

**g) Consultation des résultats**

Cette fonctionnalité permet de consulter des listes suite à la clôture d'une vente. Elle n'a donc pas d'invariant

<b>UC Consultation des résultats</b>
Spécialise : - Include - Extend
Invariant : Il faut qu'il y ait au moins une enchère à traiter
Acteur principal : Organisateur
<b>Description</b> La consultation des résultats consiste à la génération et l'affichage des listes des gagnants, des lots gagnés, des lots invendus, des lots gagnés à expertiser ...
<b>Exception</b> - Si pas d'enchère : affichage de liste(s) vide(s)

**h) Paiement des vendeurs**

<b>UC paiement des vendeurs</b>
Spécialise : Include Extend
Invariant : - Le vendeur doit exister. - Le vendeur doit avoir proposé au moins un lot gagné. - Tous ses lots gagnés doivent être réglés.
Acteur principal : Organisateur
<b>Description</b> Le paiement des vendeurs consiste à calculer les montants à payer aux vendeurs et à les stocker : création objets « ReglementVente».
<b>Exception</b> - Le vendeur doit exister : envoi d'un message d'erreur et annulation du traitement. - Le vendeur doit avoir proposé au moins un lot gagné : avertir l'utilisateur et annuler le traitement. - Tous ses lots gagnés doivent être réglés : on avertit l'utilisateur et on annule l'opération.

**i) Mise à jour des lots**

<b>UC mise à jour des lots</b>
Spécialise : <ul style="list-style-type: none"> <li>- Include</li> <li>- Extend</li> </ul>
Invariant <ul style="list-style-type: none"> <li>- Lot à supprimer ou à modifier doit exister.</li> <li>- Le lot à ajouter ne doit pas exister.</li> <li>- Les saisies doivent être correctes</li> </ul>
<b>Description</b> La mise à jour des lots comprend : <ul style="list-style-type: none"> <li>- Ajouter un lot</li> <li>- Supprimer un lot</li> <li>- Modifier état du lot</li> </ul> Cas : Ajouter un lot <ul style="list-style-type: none"> <li>• Saisie des infos du lot</li> </ul> Cas : Modifier un lot <ul style="list-style-type: none"> <li>• Saisir les infos à modifier</li> </ul> Cas : Supprimer un lot <ul style="list-style-type: none"> <li>• Saisir le numéro de lot à supprimer.</li> </ul>
<b>Exception</b> <ul style="list-style-type: none"> <li>- Lot à supprimer ou à modifier doit exister : envoyer un message d'erreur et interrompre le traitement.</li> <li>- Le lot à ajouter ne doit pas exister : envoyer un message d'erreur et interrompre le traitement.</li> <li>- Les saisies doivent être correctes : envoyer un message d'erreur et interrompre le traitement.</li> </ul>

**j) Mise à jour des personnes**

<b>UC mise à jour des lots</b>
Spécialise :
Invariant <ul style="list-style-type: none"> <li>- La personne à supprimer ou à modifier doit exister.</li> <li>- La personne à ajouter ne doit pas exister.</li> <li>- Les saisies doivent être correctes</li> </ul>
<b>Description</b> La mise à jour d'une personne comprend : <ul style="list-style-type: none"> <li>- L'ajout.</li> <li>- La suppression.</li> <li>- La modification</li> </ul> Cas : Ajout <ul style="list-style-type: none"> <li>• Saisie des infos sur la personne</li> </ul> Cas : Modification <ul style="list-style-type: none"> <li>• Saisir les infos à modifier</li> </ul> Cas : Suppression <ul style="list-style-type: none"> <li>• Saisir le numéro de la personne à supprimer.</li> </ul>
<b>Exception</b> <ul style="list-style-type: none"> <li>- La personne à supprimer ou à modifier doit exister : envoyer un message d'erreur et interrompre le traitement.</li> <li>- La personne à ajouter ne doit pas exister : envoyer un message d'erreur et interrompre le traitement.</li> <li>- Les saisies doivent être correctes : envoyer un message d'erreur et interrompre le traitement.</li> </ul>

**k) Mise à jour des experts**

<b>UC mise à jour des lots</b>
Spécialise : Mise à jour des personnes
Invariant Hérite de l'invariant de Mise à jour Personne.
<b>Description</b> Même chose mais pour l'expert avec la compétence à gérer en plus.
<b>Exception</b> - Hérite des exceptions de Mise à jour Personne.

**l) Mise à jour des acheteurs**

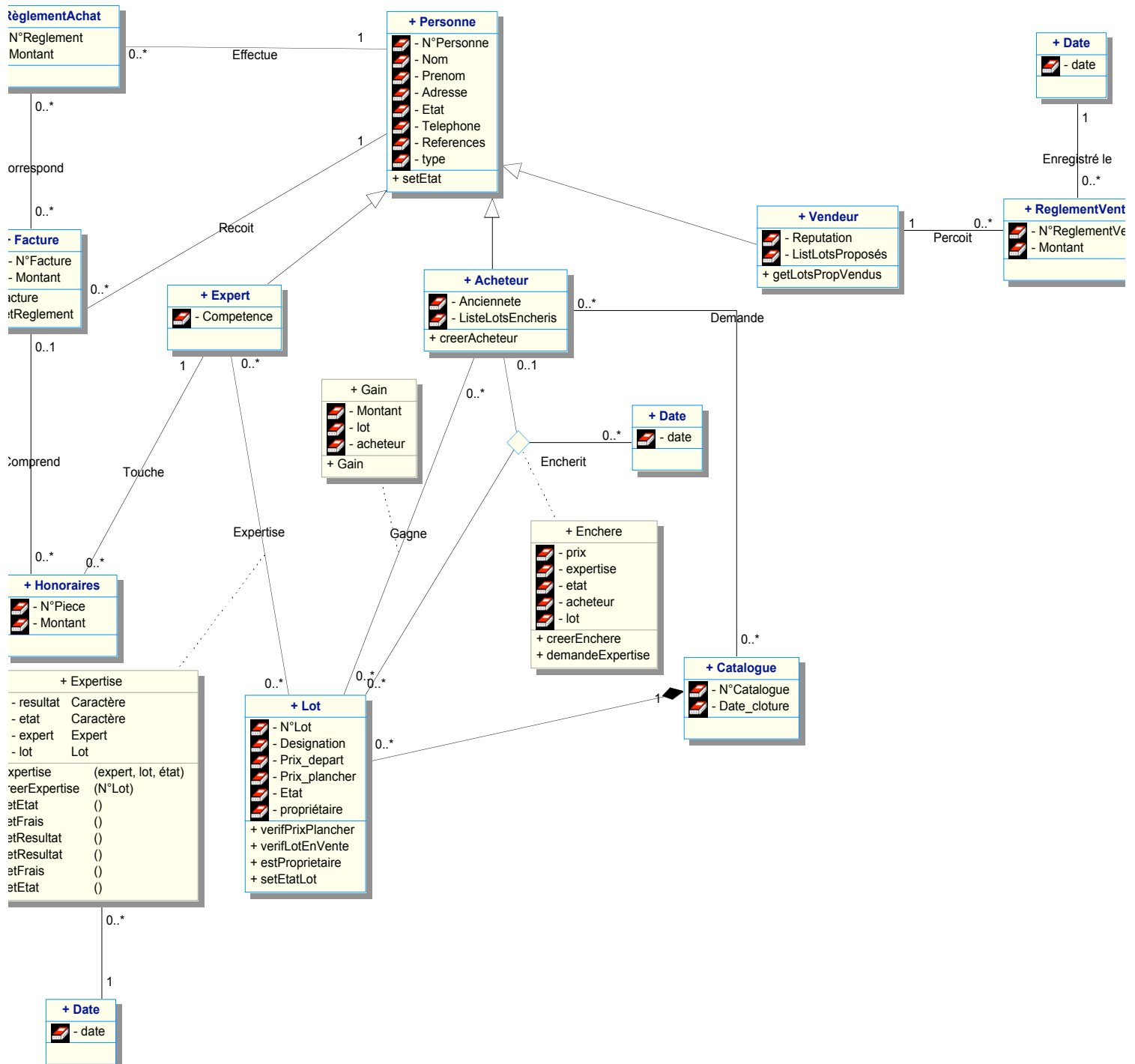
<b>UC mise à jour des lots</b>
Spécialise : Mise à jour des personnes
Invariant Hérite de l'invariant de Mise à jour Personne.
<b>Description</b> Même chose mais pour l'acheteur avec l'ancienneté à gérer en plus.
<b>Exception</b> - Hérite des exceptions de Mise à jour Personne.

**m) Mise à jour des vendeurs**

<b>UC mise à jour des lots</b>
Spécialise : Mise à jour des personnes
Invariant Hérite de l'invariant de Mise à jour Personne.
<b>Description</b> Même chose mais pour l'expert avec la compétence à gérer en plus.
<b>Exception</b> - Hérite des exceptions de Mise à jour Personne.

### B. Diagrammes de classes

Le diagramme des classes a été mis à jour au fur et à mesure des diagrammes de séquences. Nous y ajoutons les méthodes correspondant aux messages entre les classes.



Les états possibles :

**Personne** : mauvais payeur, bon payeur, ancien, nouveau.

**Lot** : vendu, retiré, proposé, gagné, en vente.

**Enchère** : gagnante, perdante.

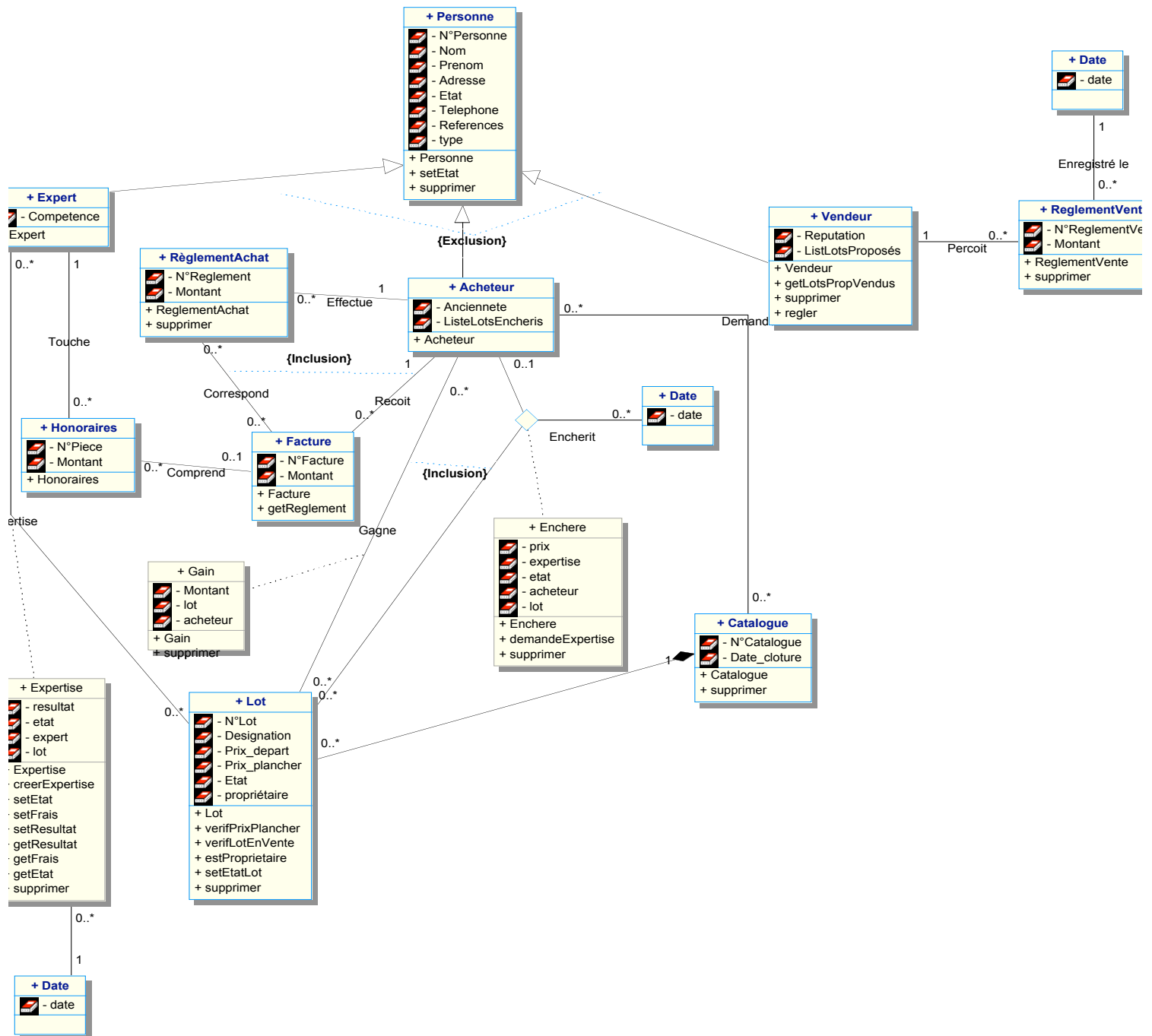
**Expertise** : ancienne, nouvelle, en cours.

Ce premier diagramme de classes ne portait pas les contraintes. Nous avons relevé les infos qu'il faut modéliser et la classe « ReglementAchat » était reliée à « Personne ». Or toute personne ne règle pas d'achats ; en effet seul un acheteur peut régler un achat. Même s'il est vrai que la classe « Acheteur » hérite de toutes les propriétés de « Personne » ; il est important de modéliser cette particularité de l'acheteur. Il en est de même pour la classe « Facture ».

Nous avons ensuite énuméré les contraintes que doit respecter le système.

- Une personne est soit un acheteur, soit un vendeur, soit un expert. En effet, nous avons décidé que lorsqu'un vendeur ou un expert souhaite acheter, il faut créer un objet « Acheteur » leur correspondant.
- Un acheteur ne peut que gagner un lot sur lequel il a enchéri.
- Un vendeur ne peut enchérir sur son propre lot.
- Les expertises nouvelles ne sont réalisées que si le lot a un acheteur.
- Le prix de départ du lot doit être toujours être supérieur au prix plancher.
- Le prix d'une offre est toujours supérieur au prix plancher.
- Deux mois avant la date de clôture, les lots non retirés doivent avoir un numéro définitif.
- Un lot non payé ne peut être « vendu ».

Nous avons donc modifié le diagramme pour obtenir ceci.



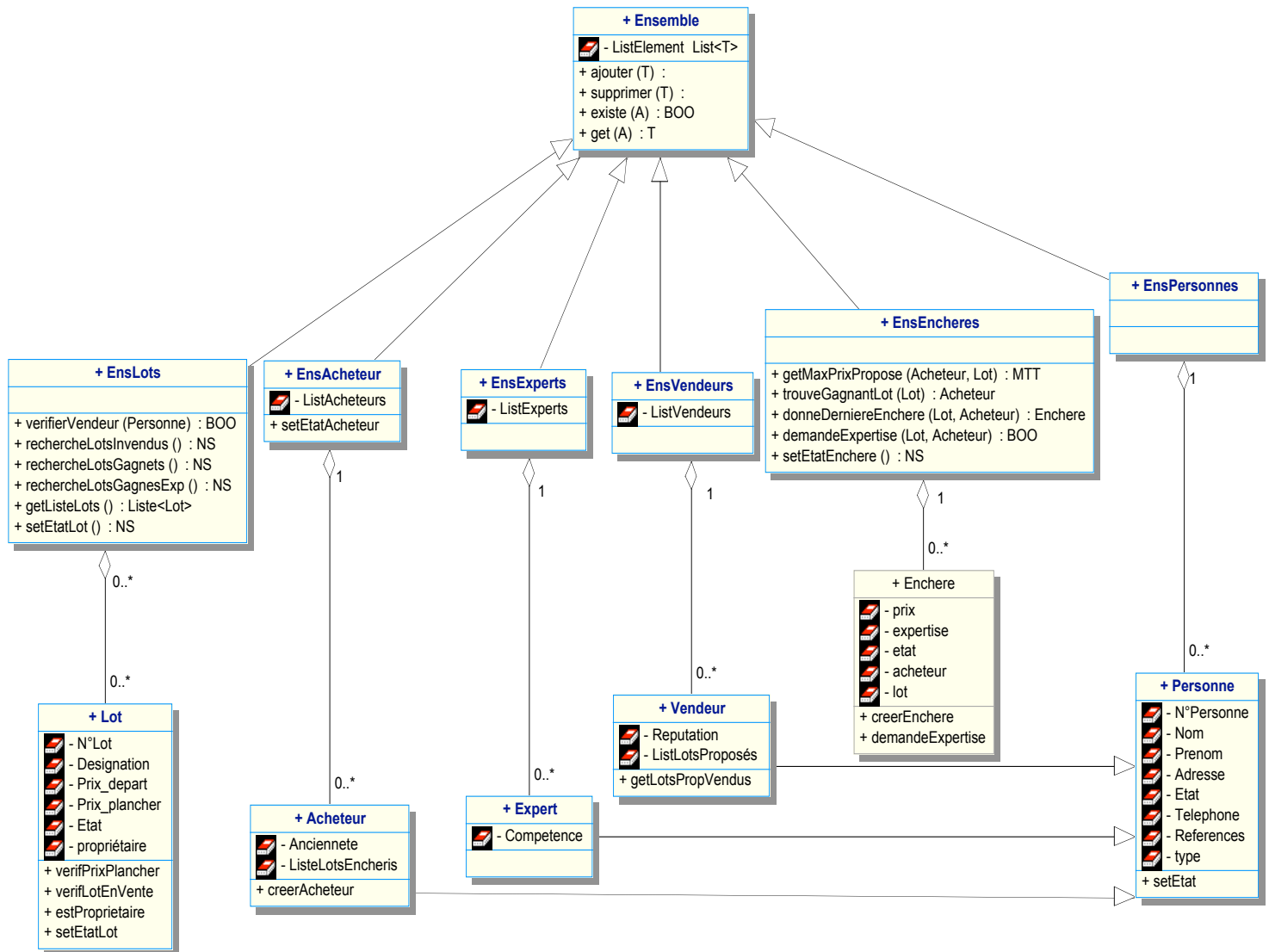
Les méthodes supprimer() permettent de détruire les objets.

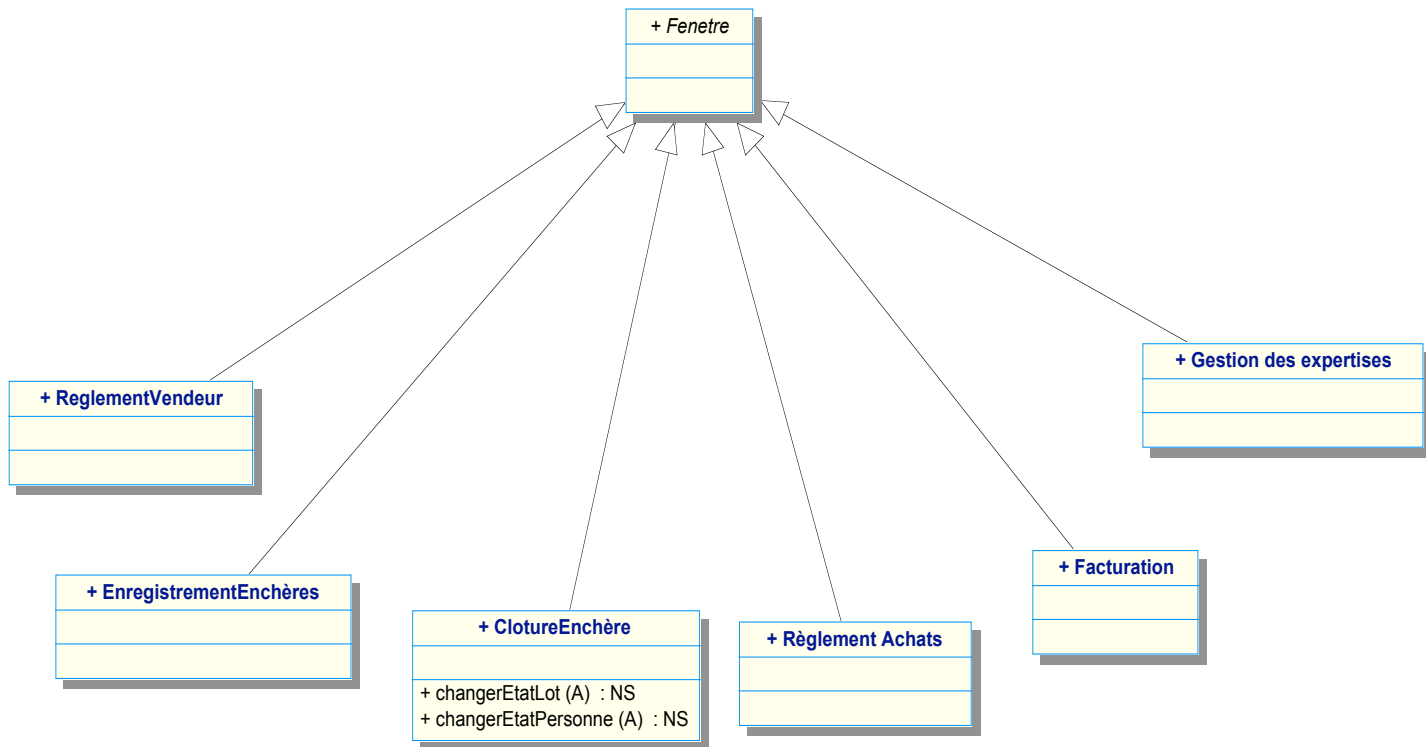
Chaque Classe a un constructeur, les sous classes de Personne ont également des constructeurs qui sont des surcharges du constructeur hérité Personne(). Nous les avons renommés pour faciliter la compréhension de nos modèles.

Nous sommes demandés maintenant si notre diagramme de classes, respecte toutes les contraintes.

Les deux premières contraintes sont formalisées sur le diagramme des classes, les autres seront exprimées en langage OCL.



**DIAGRAMME DES ENSEMBLES**

**DIAGRAMME IHM****C. Diagrammes de scénarios, séquences et collaboration**

Les différents scénarios et diagrammes de séquences découlent des cas d'utilisation. En effet, ils sont fonction du nombre de cas d'utilisation et des exceptions de chacun des cas.

**1. Gestion des ventes**

Cette fonction permet d'enregistrer les différentes offres des acheteurs, de clôturer la vente et d'éditer les résultats.

**a) Saisie des enchères**

L'application doit gérer huit situations différentes, celles relevées dans la description de ce cas.

Les étapes principales du scénario sont les suivantes :

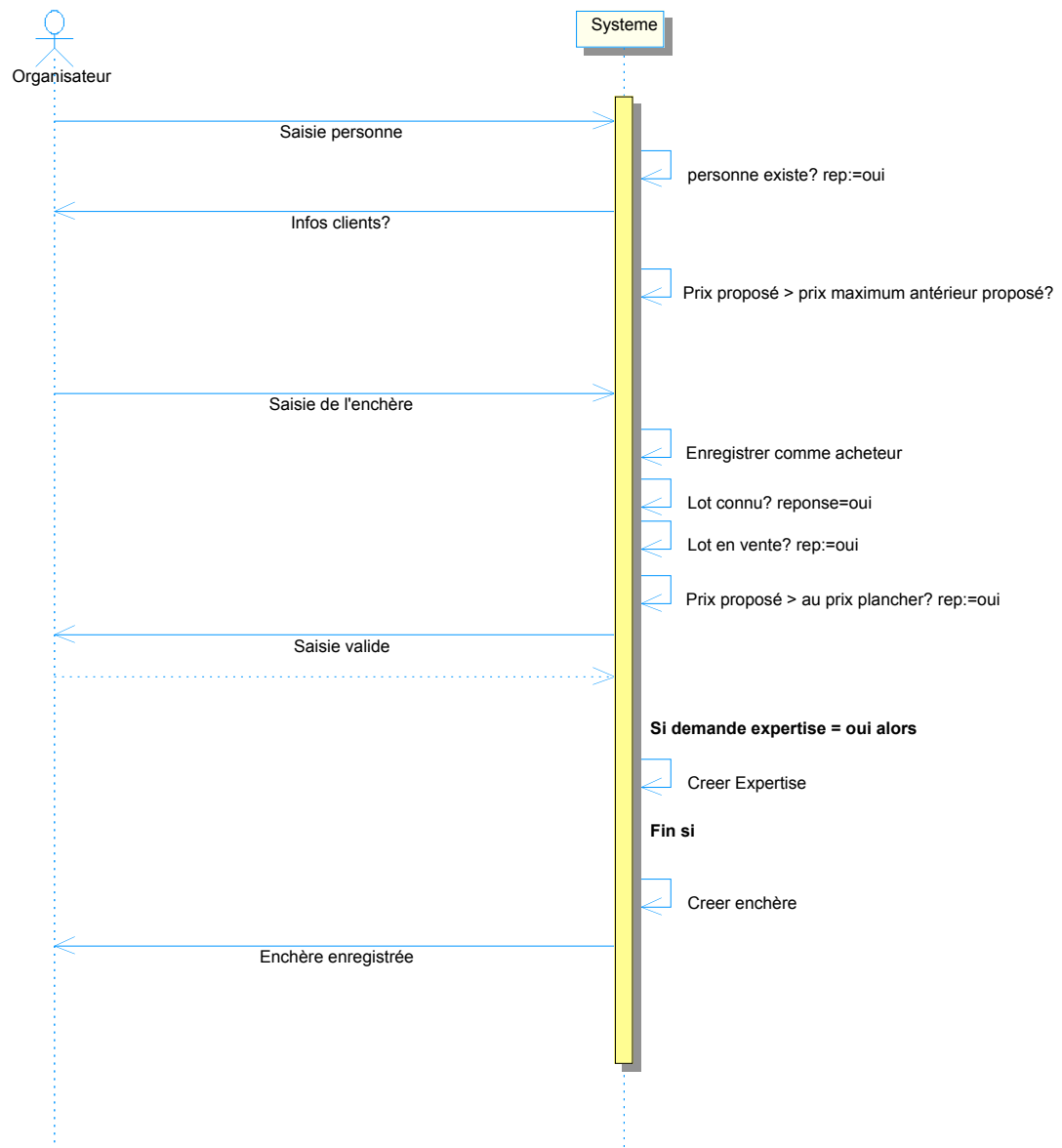
- Saisie du numéro de la personne.
- Saisie du numéro de lot et du prix de l'offre.
- Saisie de la demande d'expertise.
- Enregistrement de l'offre.

Lors de ces différentes étapes, le système doit effectuer des contrôles pour respecter l'invariant lié à ce traitement.

**Diagrammes de scénario et de séquences et collaboration**

- Personne inconnu mais n'est pas le vendeur du lot

Enregistrement des enchères : personne inconnue et type<>vendeur		
Projet : UML		
Version : 1	Créé le : 29/10/2006	Modifié le : 4/11/2006



## Enregistrement des enchères : personne inconnue et type&lt;&gt;vendeur

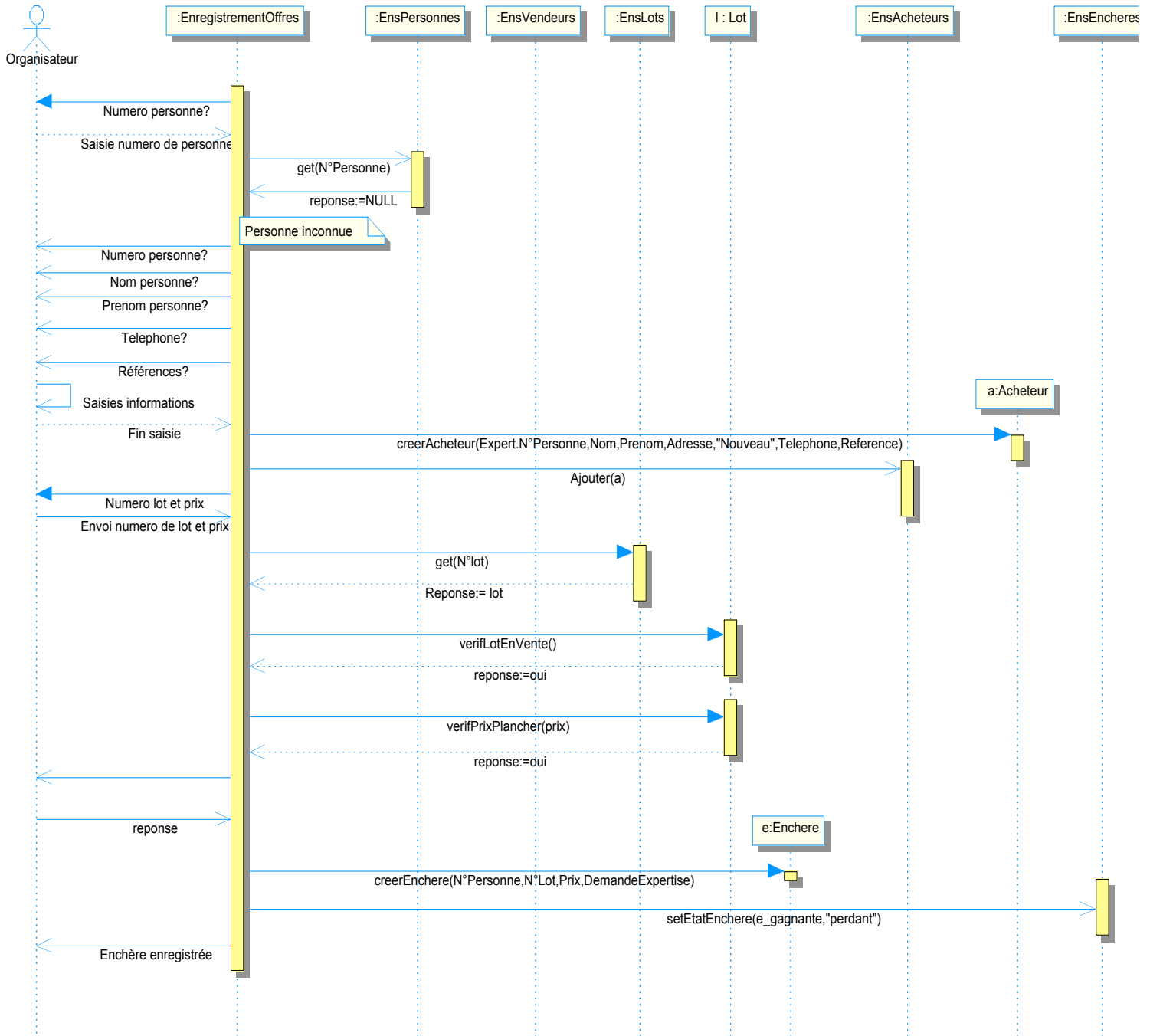
Objet : UML

Auteur : FAGBEGNON

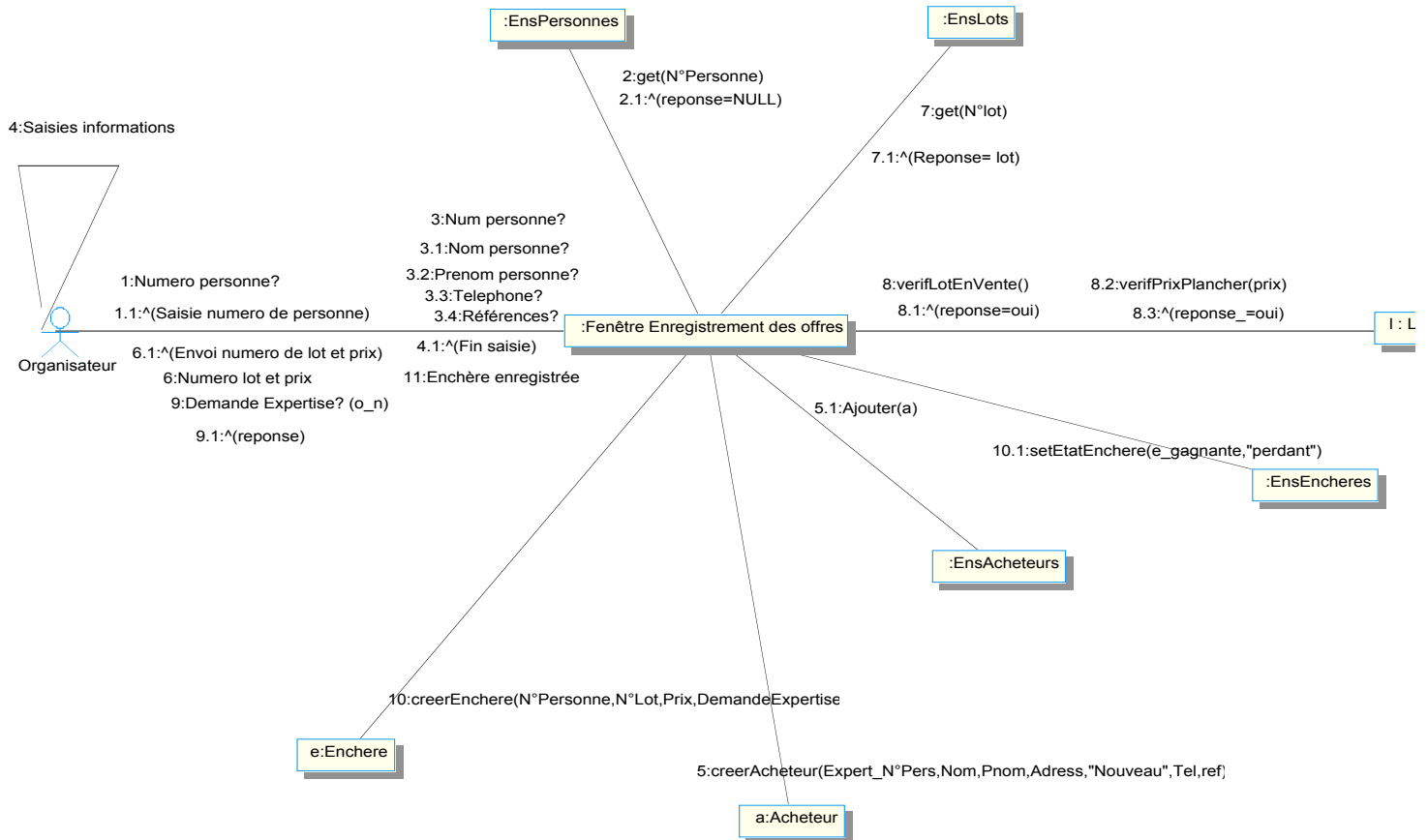
Version : 1

Créé le : 21/10/2006

Modifié le : 7/11/2006

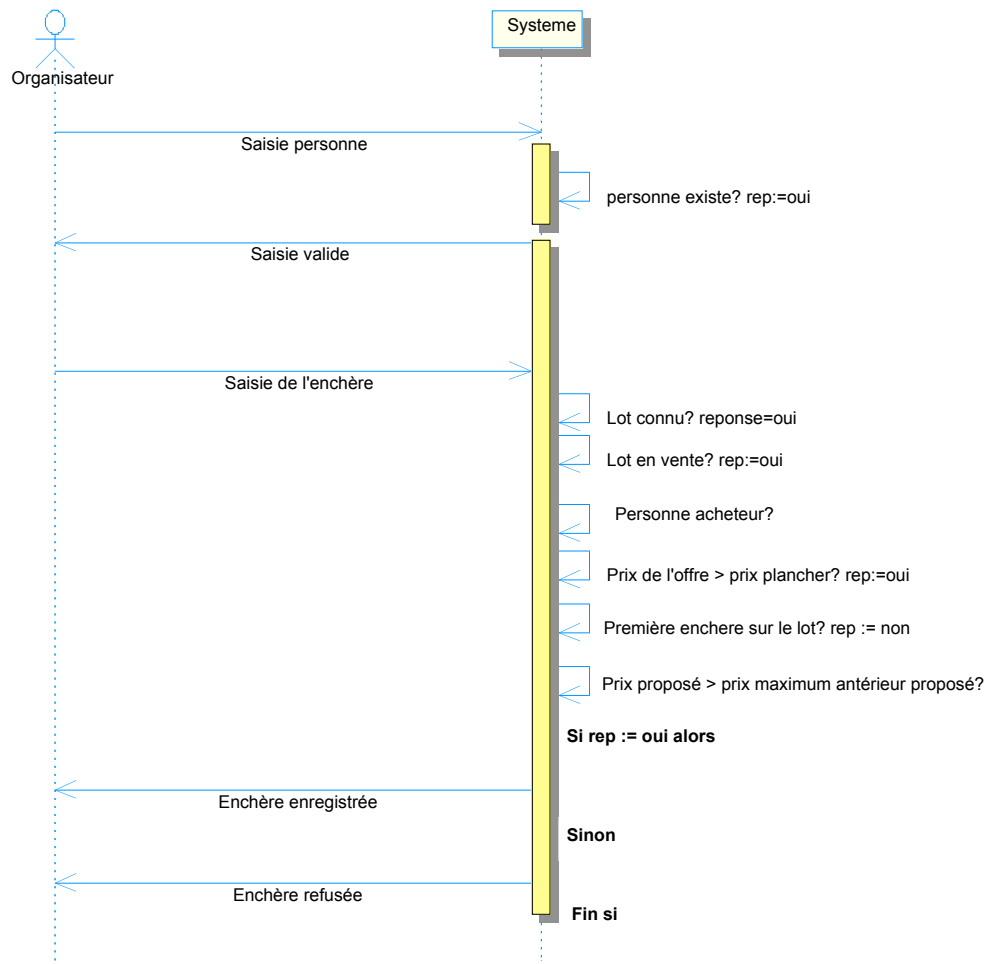


Sous-modèle1		
Titre : UML		
Version : I012		
Version : 1	Créé le : 7/11/2006	Modifié le : 8/11/2006

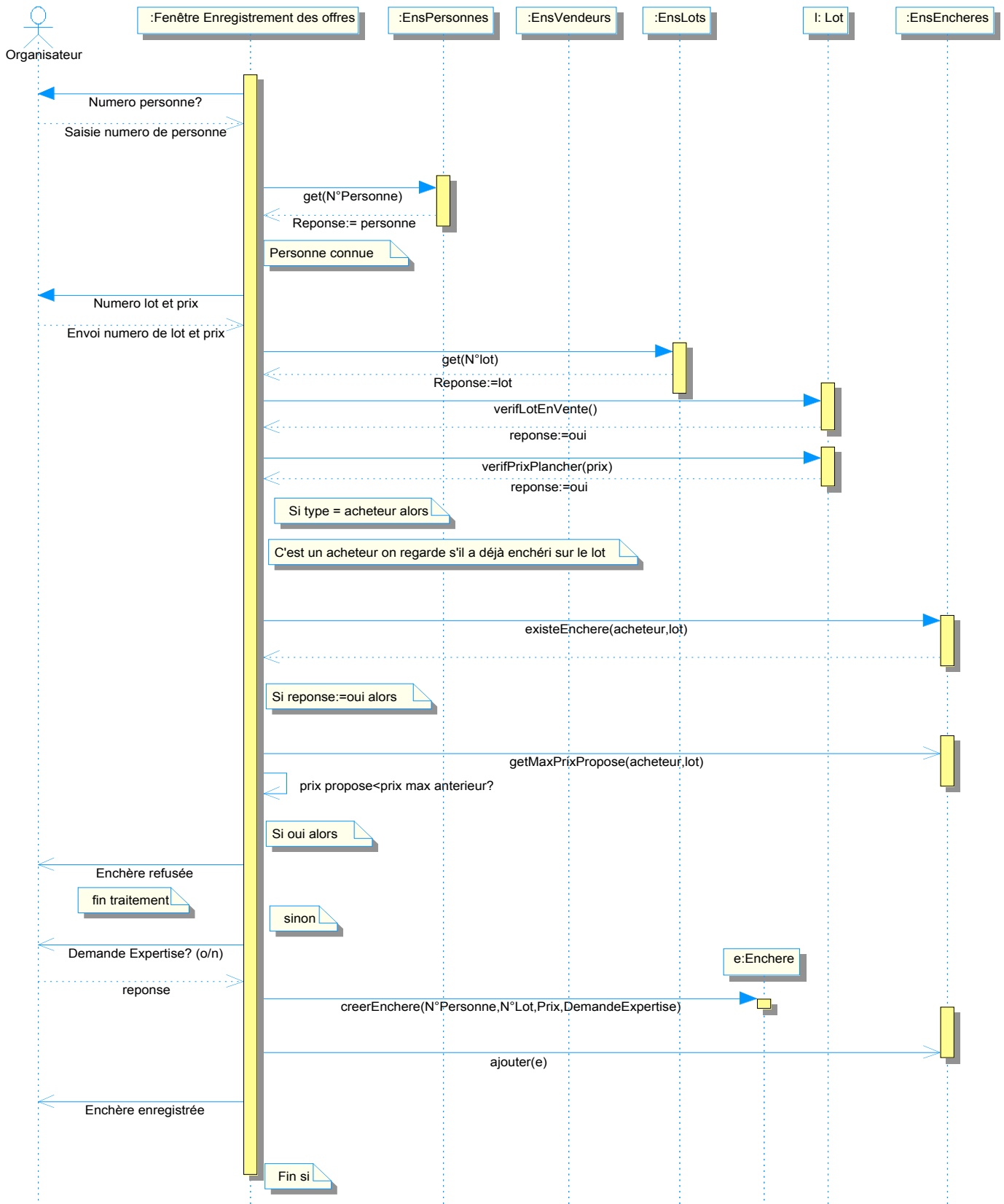


- Acheteur qui a déjà effectué une enchère sur le lot

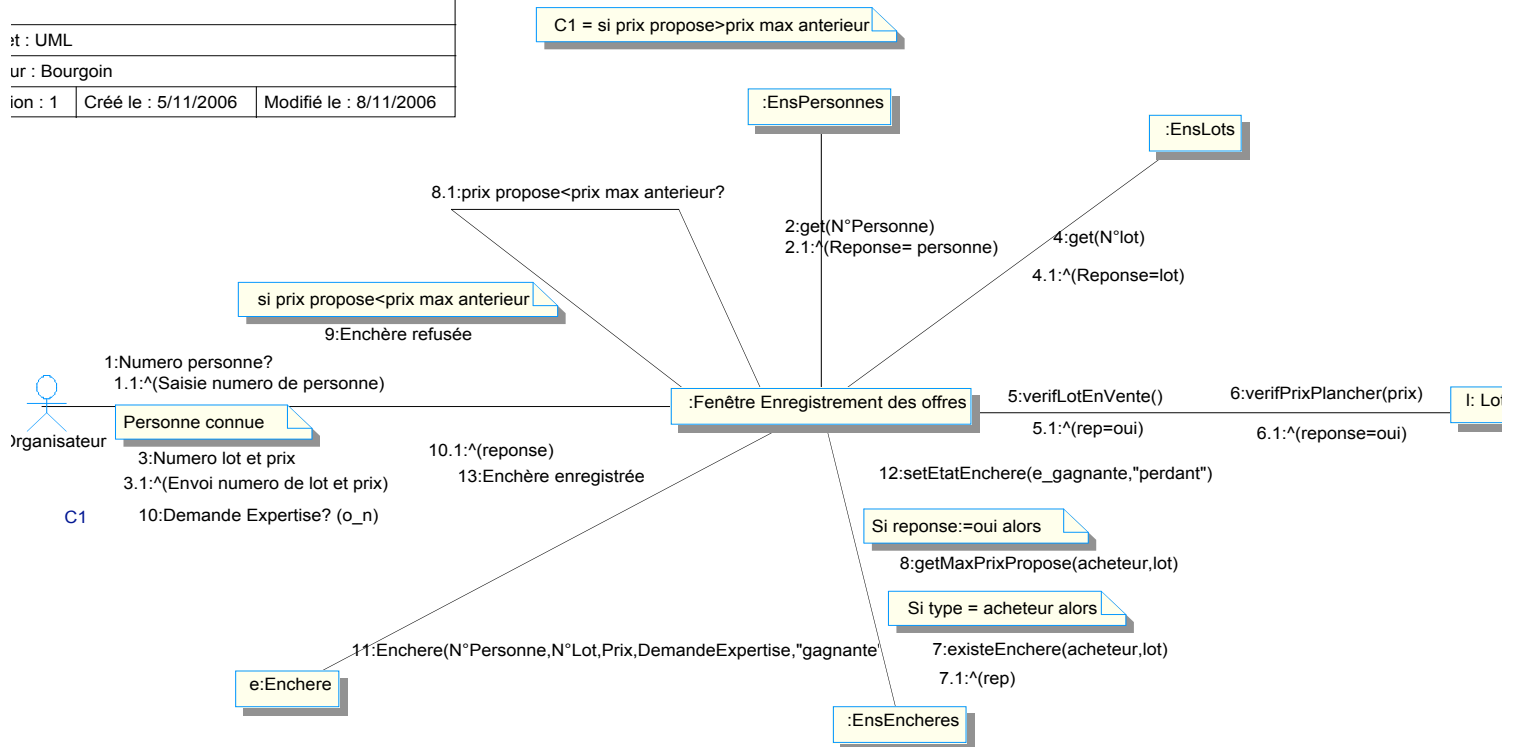
Enregistrement des offres: client acheteur et a déjà effectue une enchère sur le lot		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 29/10/2006	Modifié le : 4/11/2006



Enregistrement des offres: personne acheteur et a déjà effectue une enchère sur le lot		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 21/10/2006	Modifié le : 4/11/2006

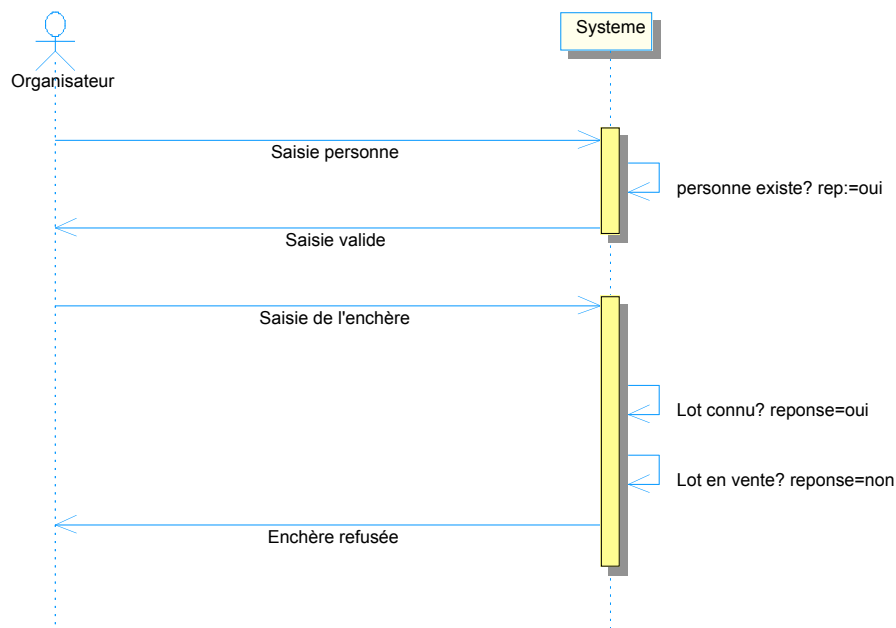


Sous-modèle1		
Titre : UML		
Auteur : Bourgoin		
Version : 1	Créé le : 5/11/2006	Modifié le : 8/11/2006



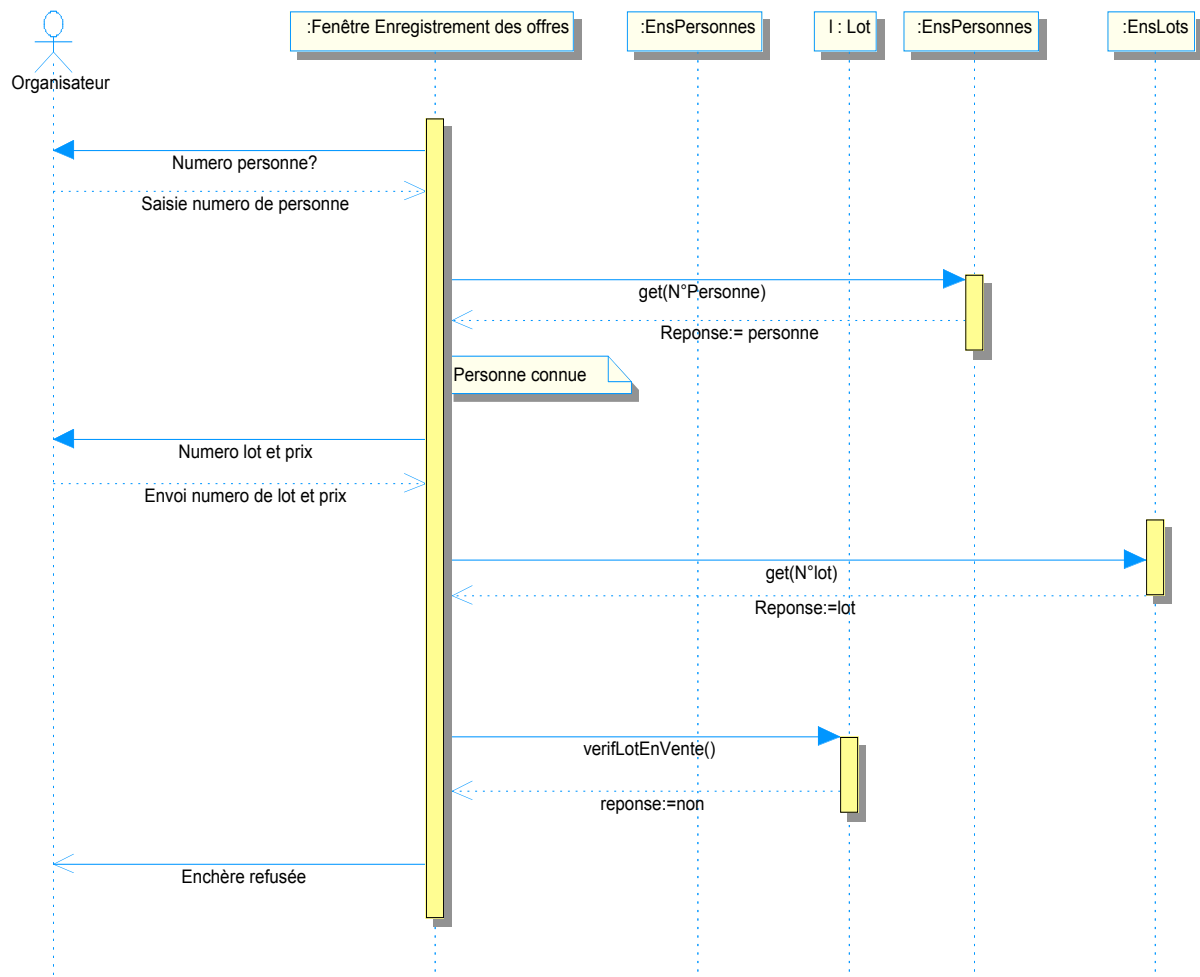
- Personne connu mais lot pas en vente

Enregistrement des offres: personne connue mais le lot n'est pas en vente		
Version : 1		
Créé le : 29/10/2006	Modifié le : 4/11/2006	

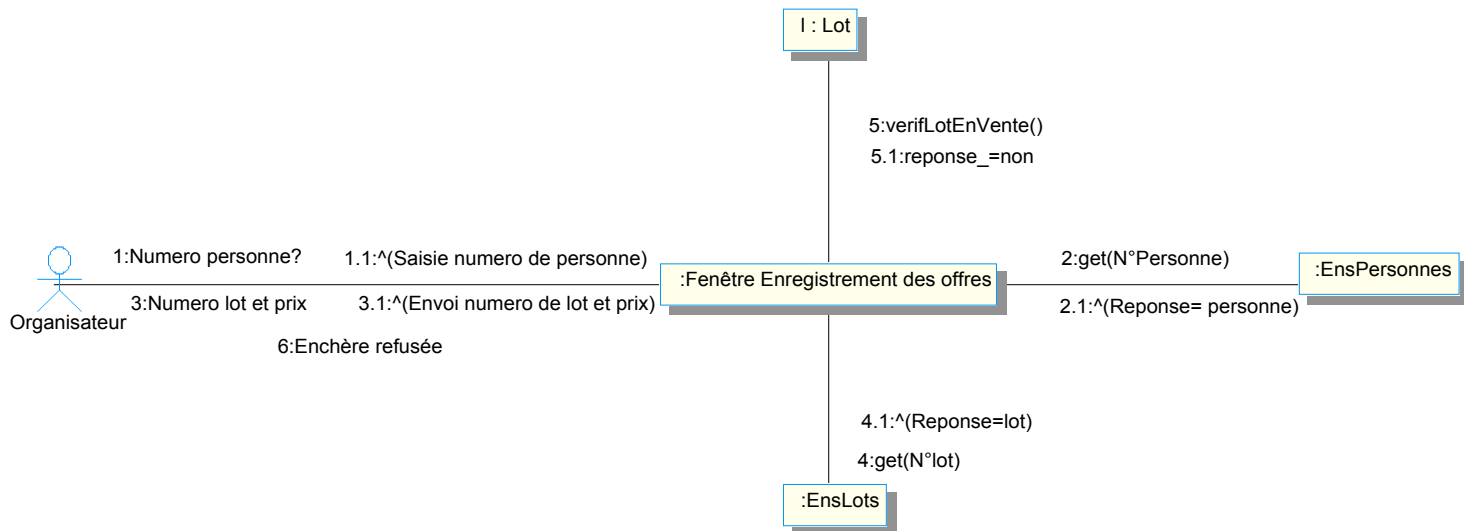




Enregistrement des offres: personne connue mais le lot n'est pas en vente		
Version : 1		
Créé le : 21/10/2006	Modifié le : 4/11/2006	

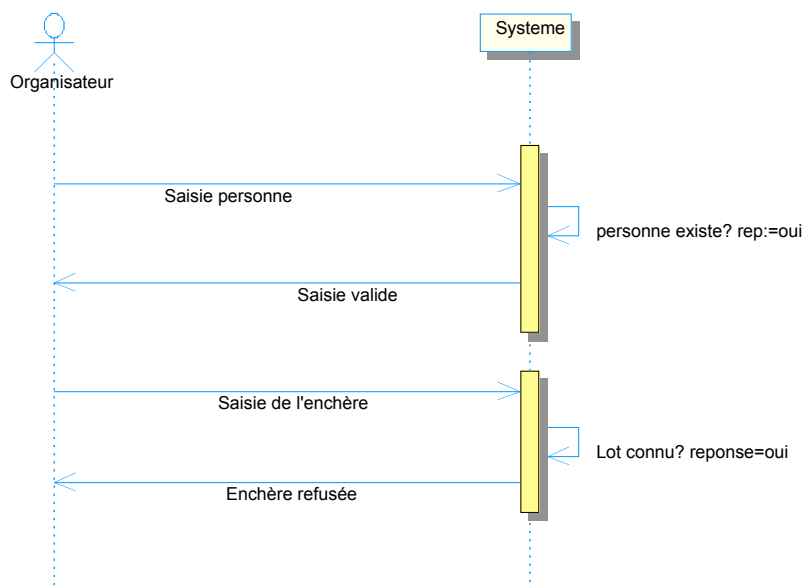


Sous-modèle1		
Projet : UML		
Auteur : Bourgoin		
Version : 1	Créé le : 5/11/2006	Modifié le : 7/11/2006

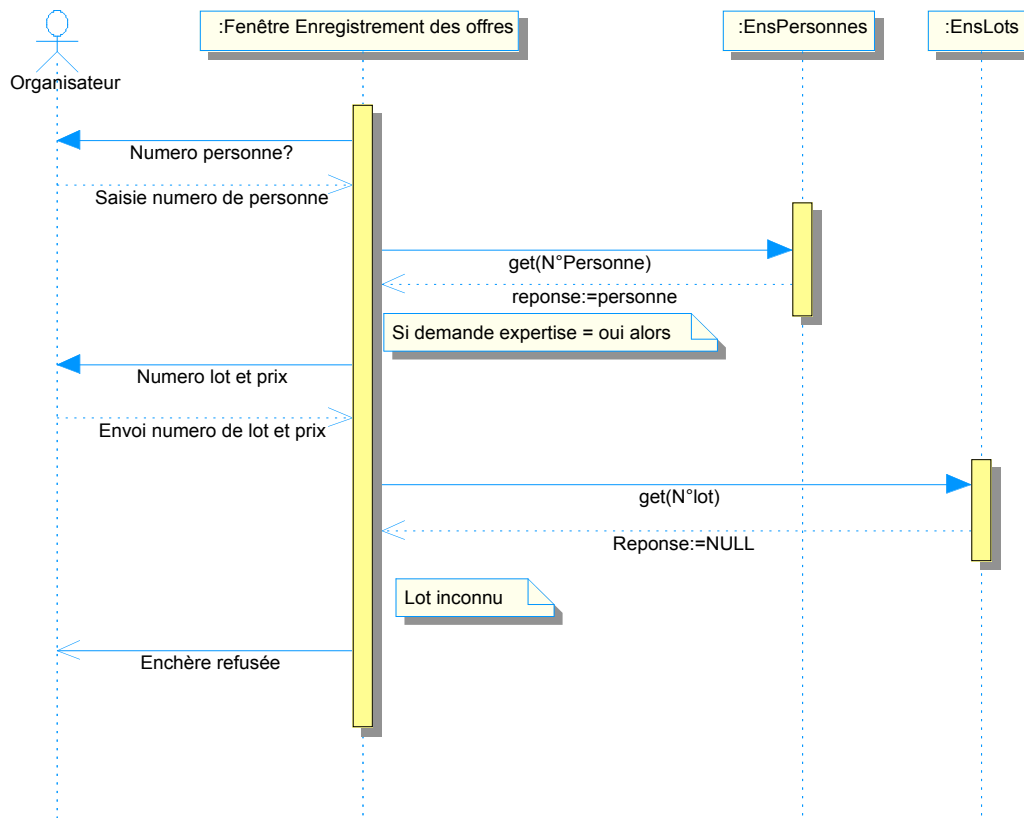


- Personne connue mais lot inconnu

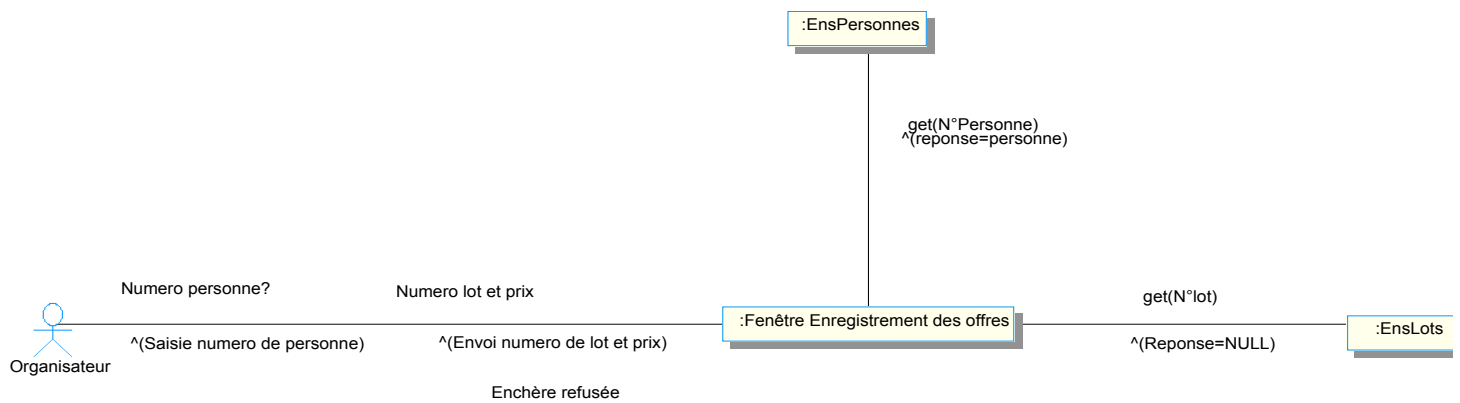
Enregistrement des offres: personne connue mais lot inconnu		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 28/10/2006	Modifié le : 4/11/2006



Enregistrement des offres: personne connue mais lot inconnu		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 21/10/2006	Modifié le : 4/11/2006

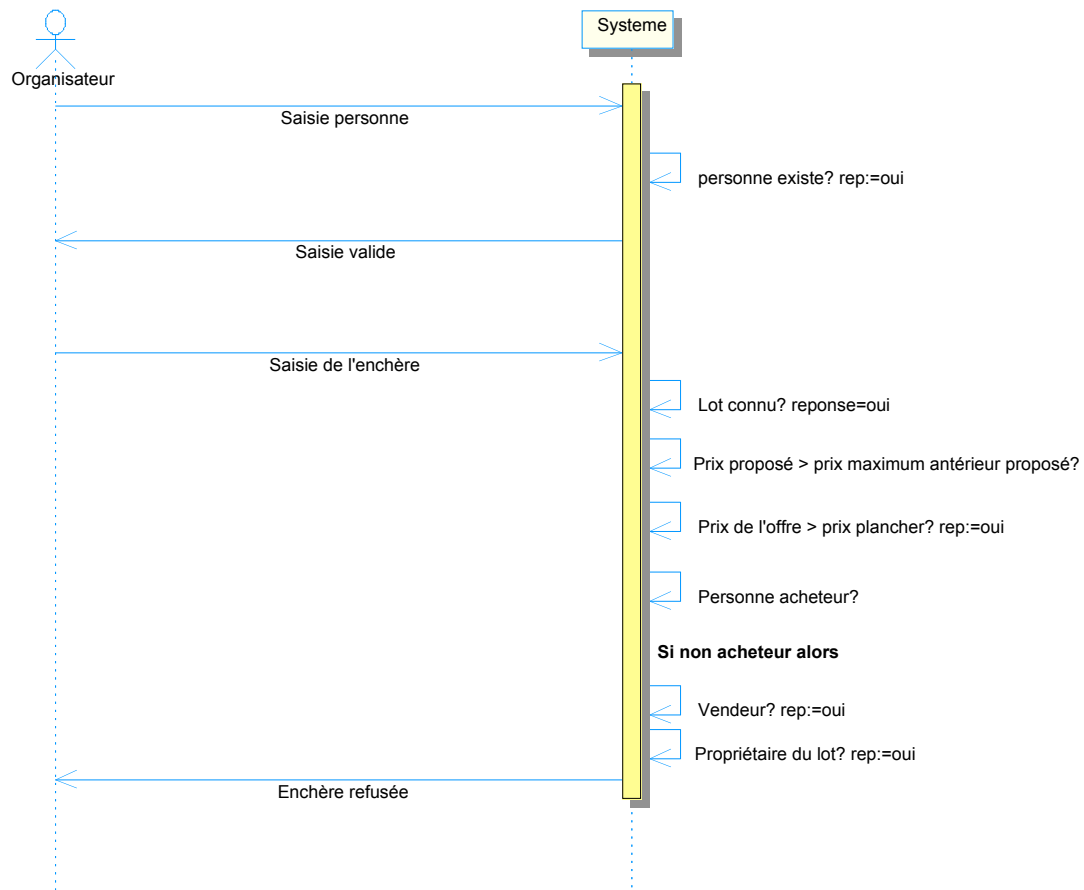


Sous-modèle1		
Projet : UML		
Auteur : I012		
Version : 1	Créé le : 7/11/2006	Modifié le : 7/11/2006

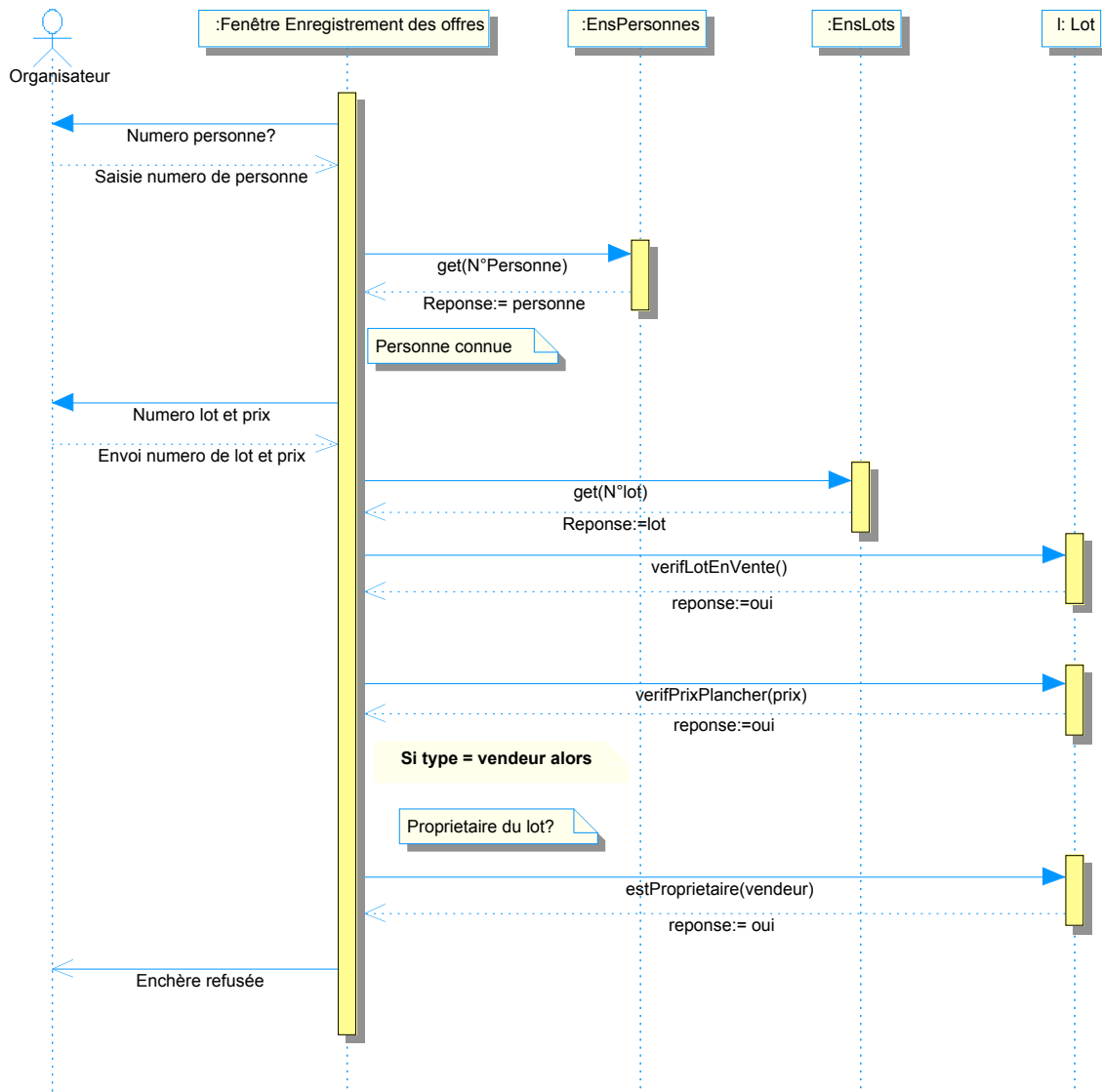


- Vendeur et propriétaire du lot

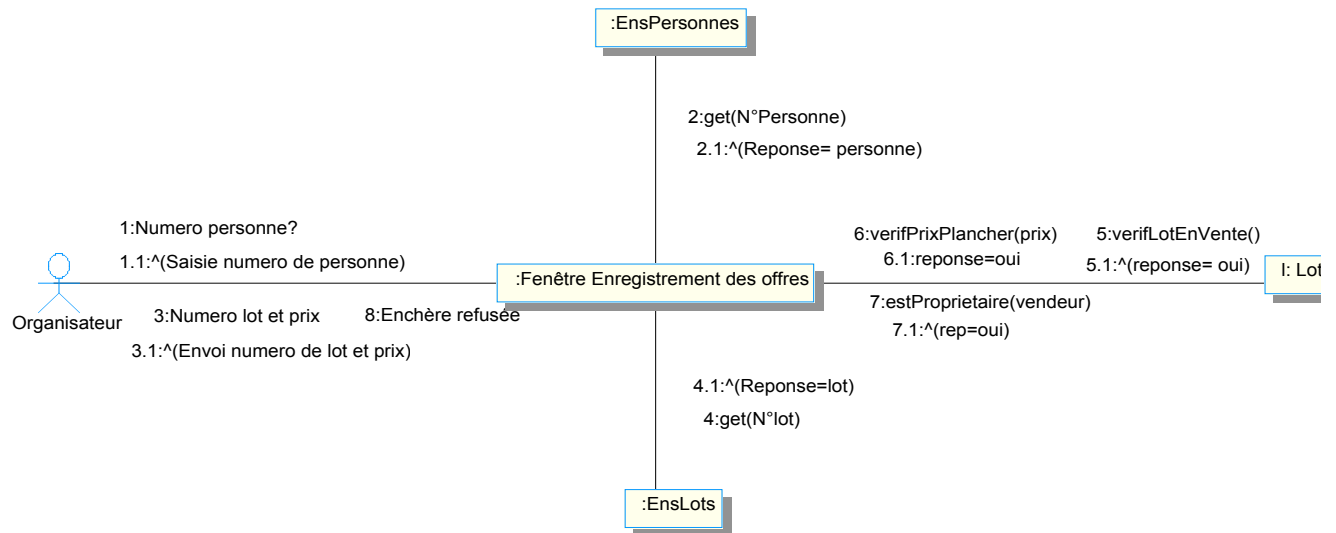
Enregistrement des offres: personne type vendeur et propriétaire du lot		
Version : 1		
Créé le : 29/10/2006		
Modifié le : 4/11/2006		



Enregistrement des offres: personne type vendeur et propriétaire du lot		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 21/10/2006	Modifié le : 4/11/2006

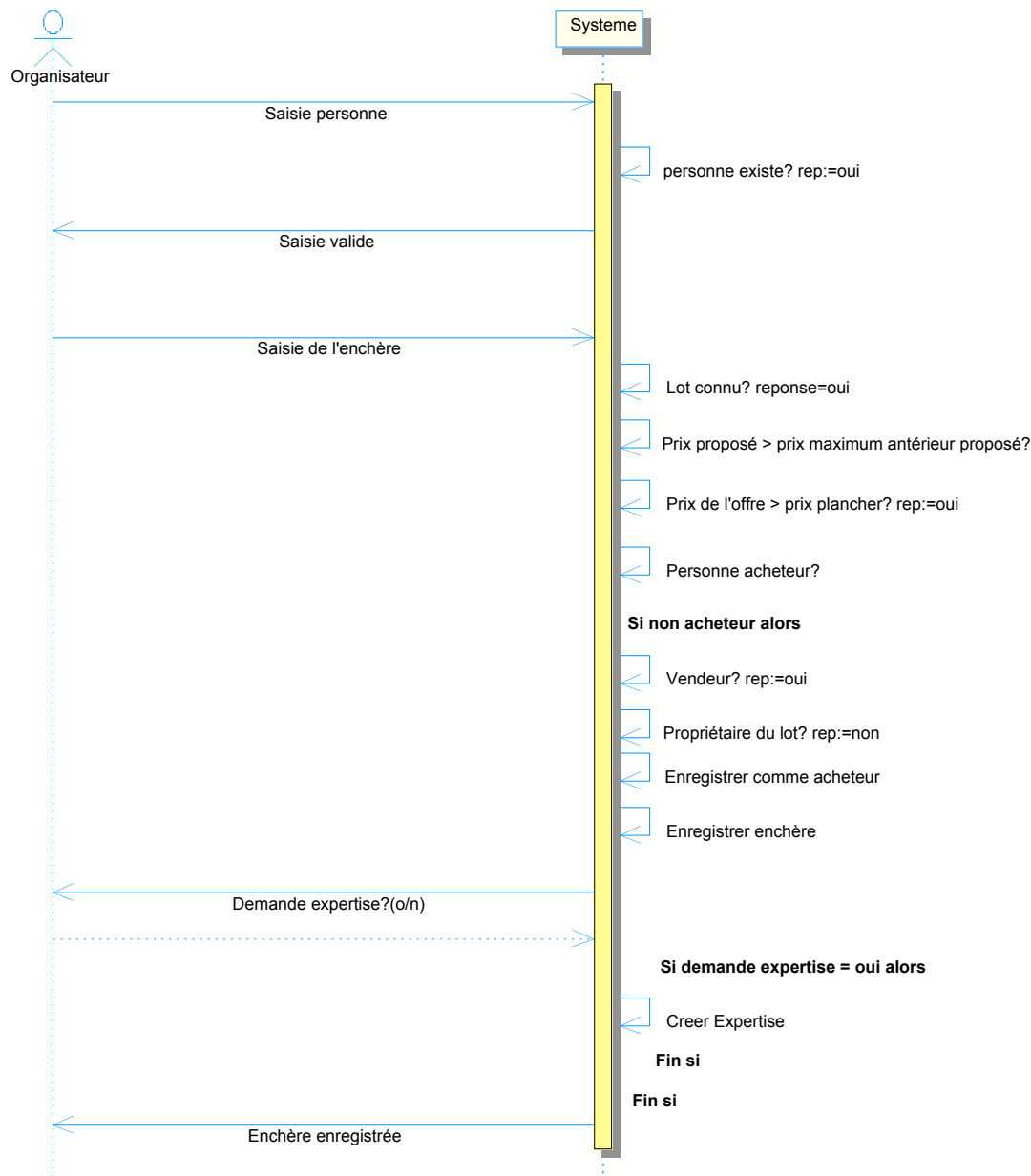


Sous-modèle1		
Projet : UML		
Auteur : Bourgoin		
Version : 1	Créé le : 5/11/2006	Modifié le : 8/11/2006



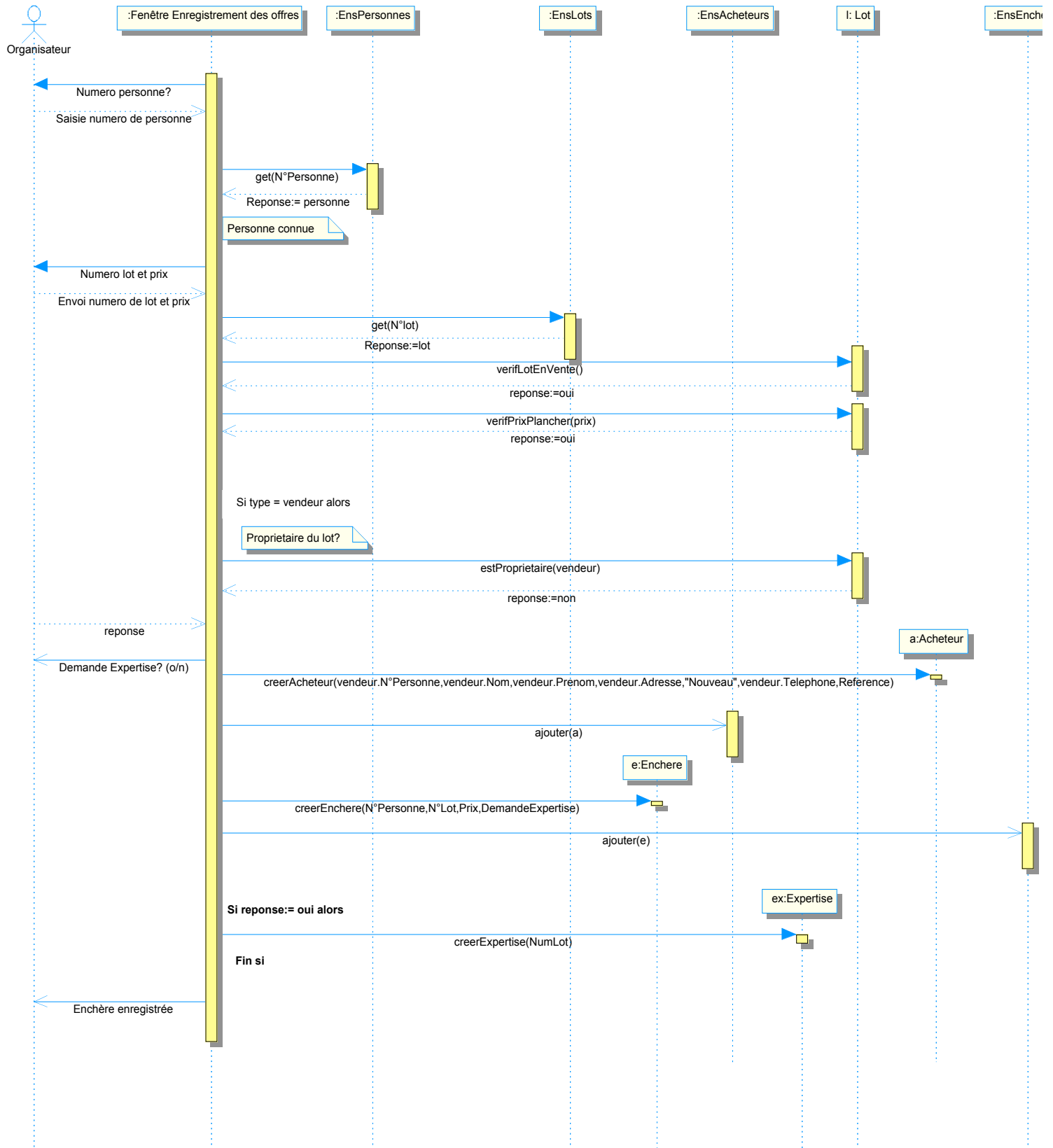
- Vendeur mais pas propriétaire du lot

Enregistrement des offres: perssone type vendeur mais pas propriétaire du lot		
Version : 1	Créé le : 29/10/2006	Modifié le : 4/11/2006

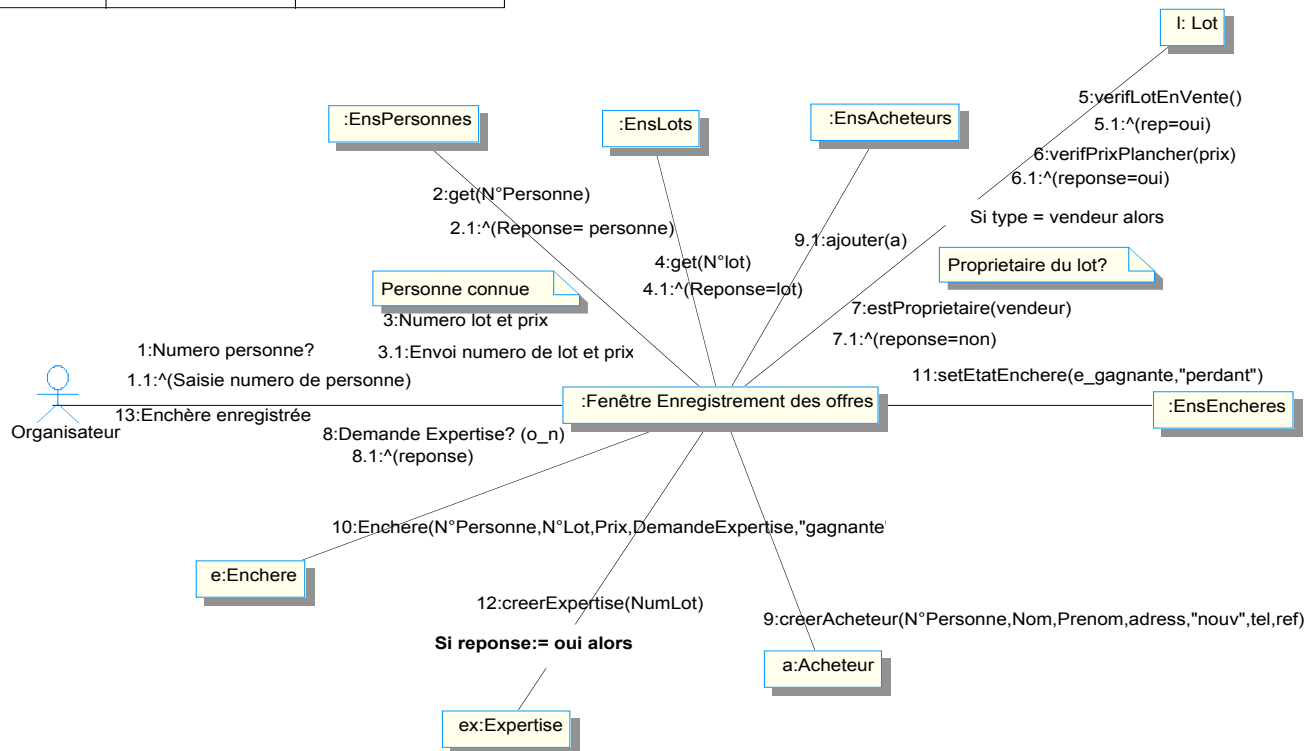




Enregistrement des offres: personne type vendeur mais pas propriétaire du lot		
objet : UML		
auteur : FAGBEGNON		
version : 1	Créé le : 21/10/2006	Modifié le : 4/11/2006

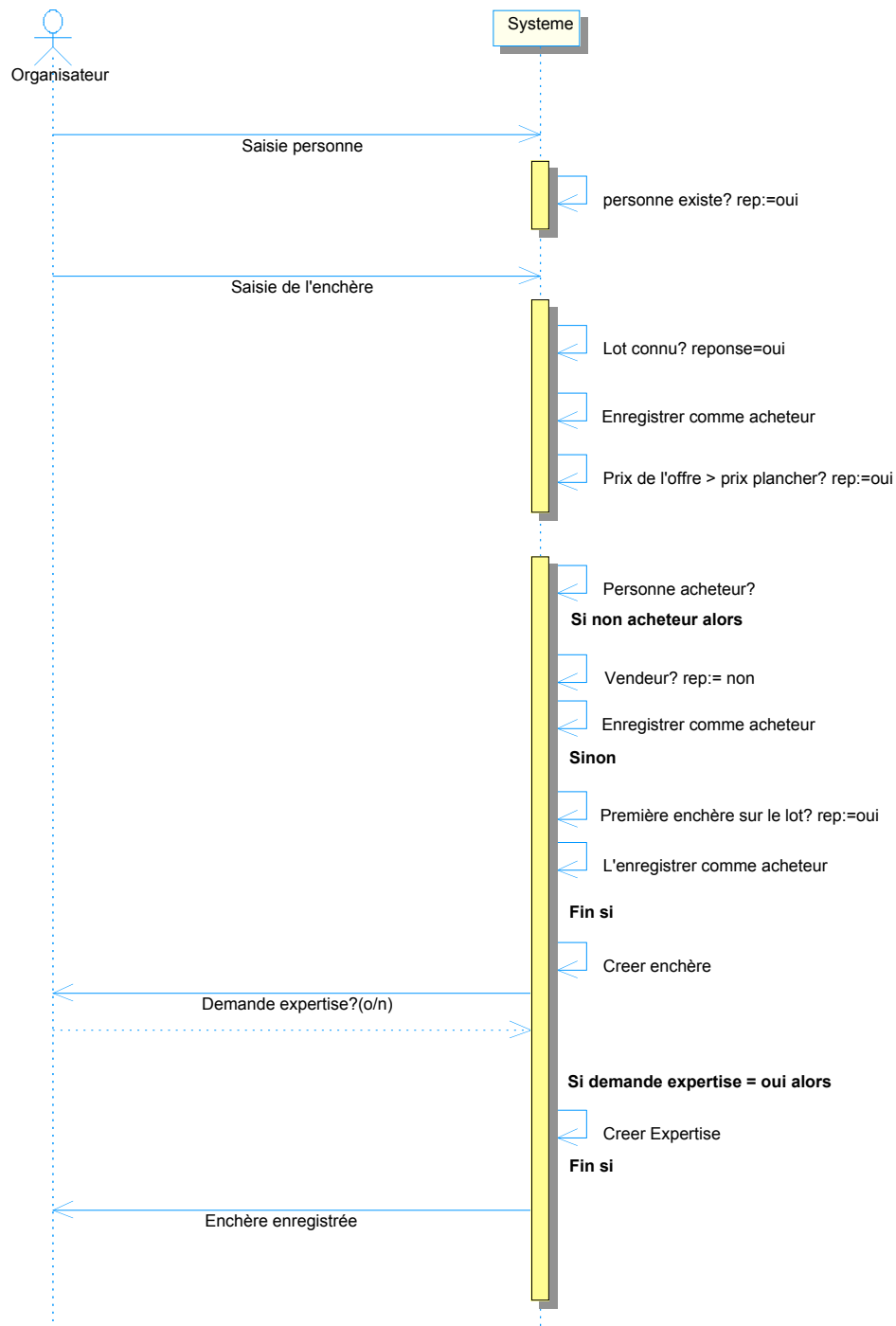


Sous-modèle1		
Projet : UML		
Auteur : Bourgoin		
Version : 1	Créé le : 5/11/2006	Modifié le : 8/11/2006

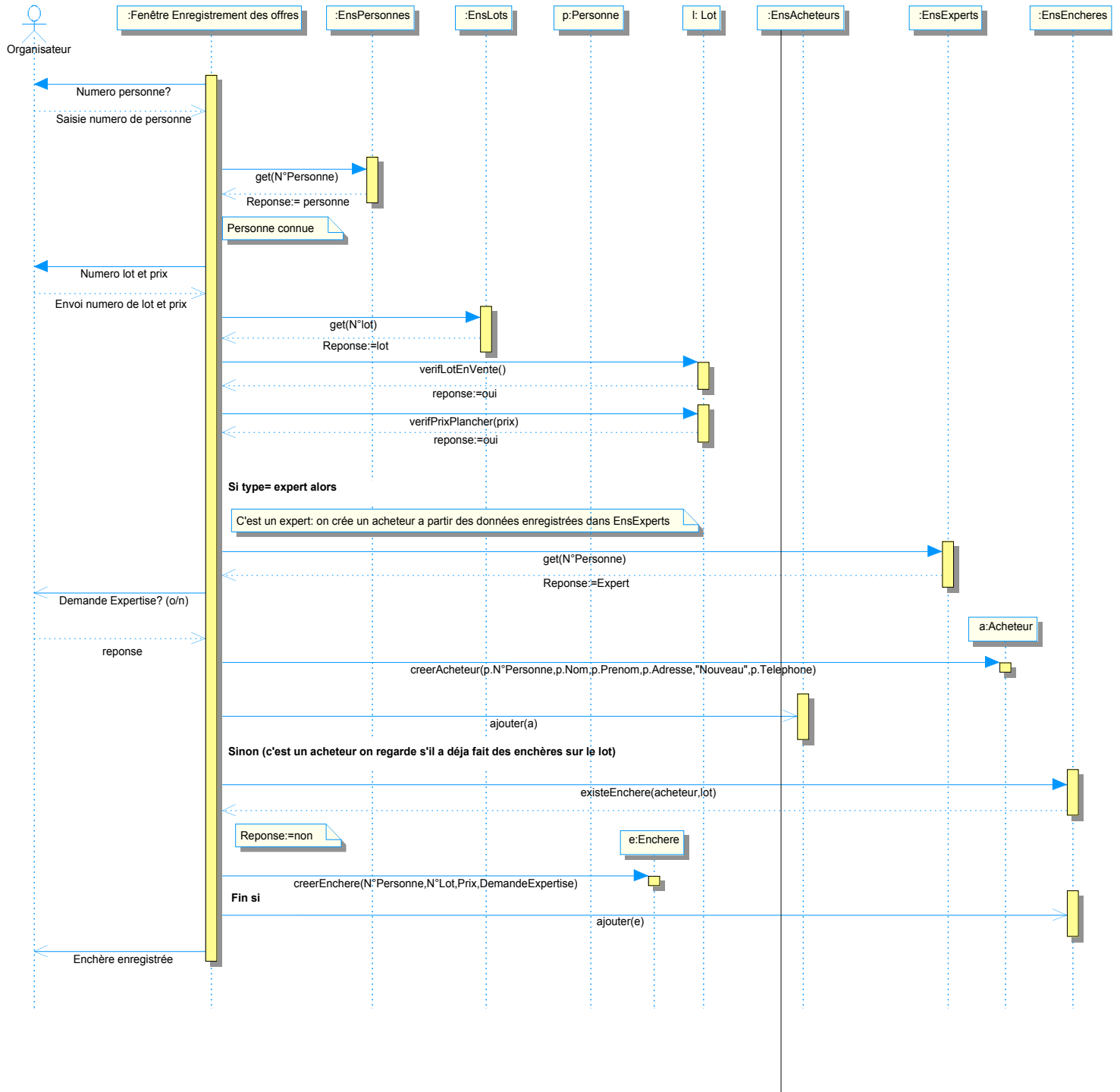


- Personne non vendeur

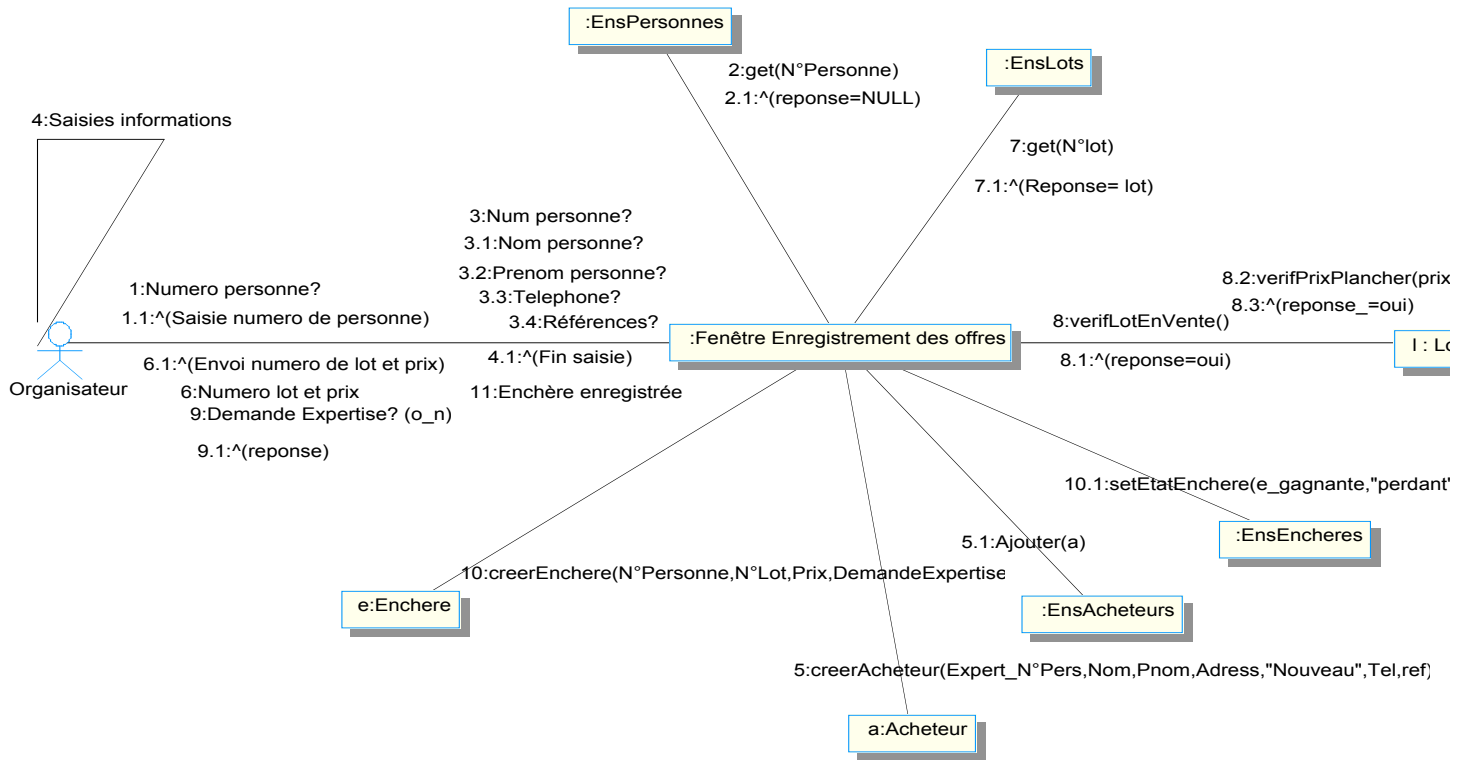
Enregistrement des offres: type personne <> vendeur		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 28/10/2006	Modifié le : 29/10/2006



Enregistrement des offres: personne connue et type <> vendeur		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 21/10/2006	Modifié le : 4/11/2006

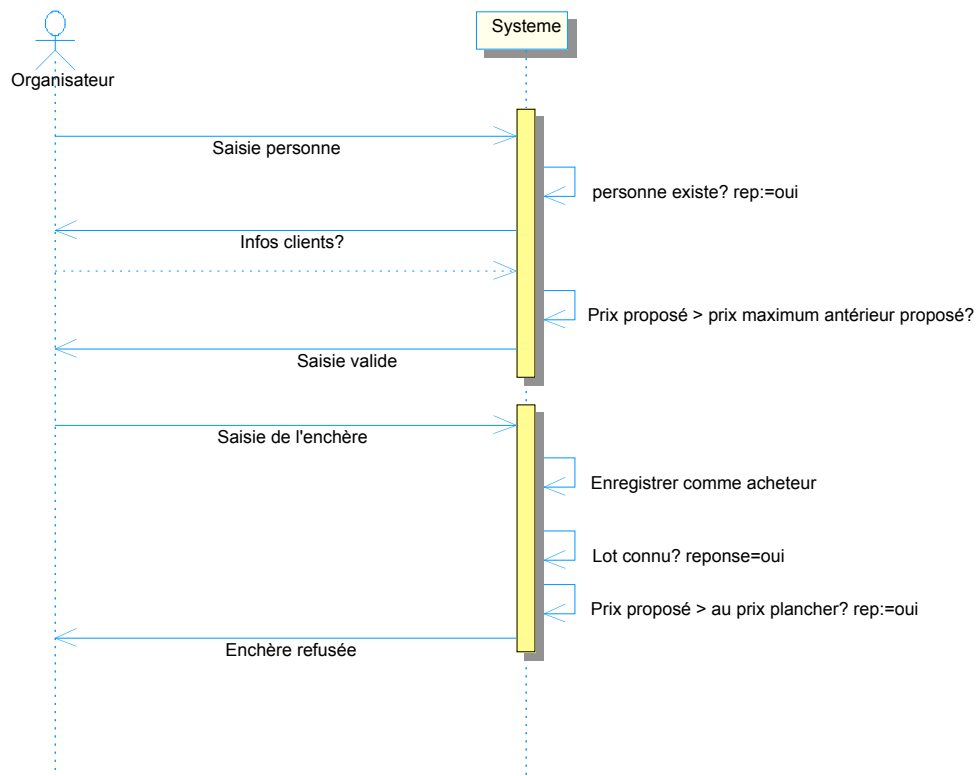


Sous-modèle1		
Projet : UML		
Auteur : I012		
Version : 1	Créé le : 7/11/2006	Modifié le : 8/11/2006

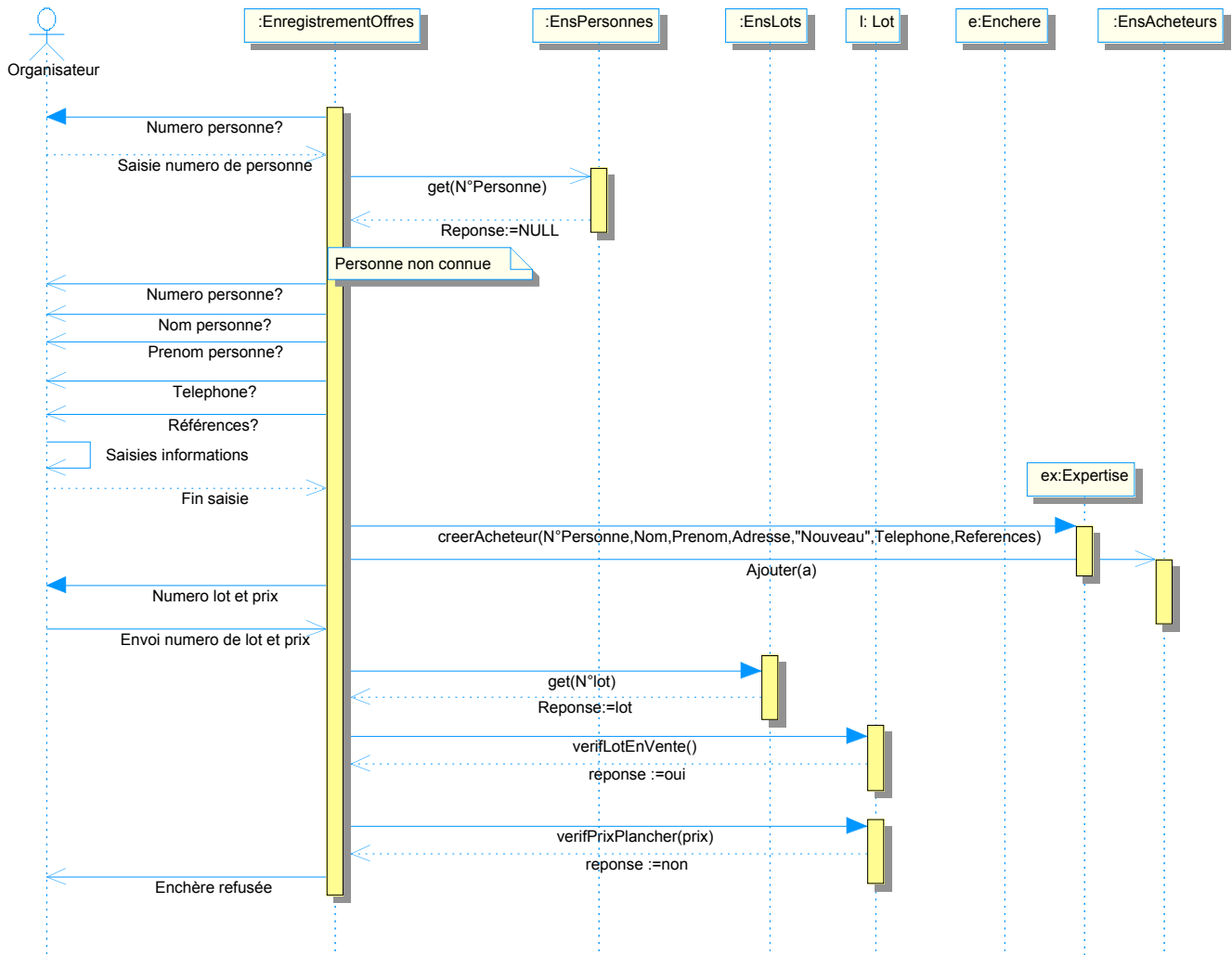


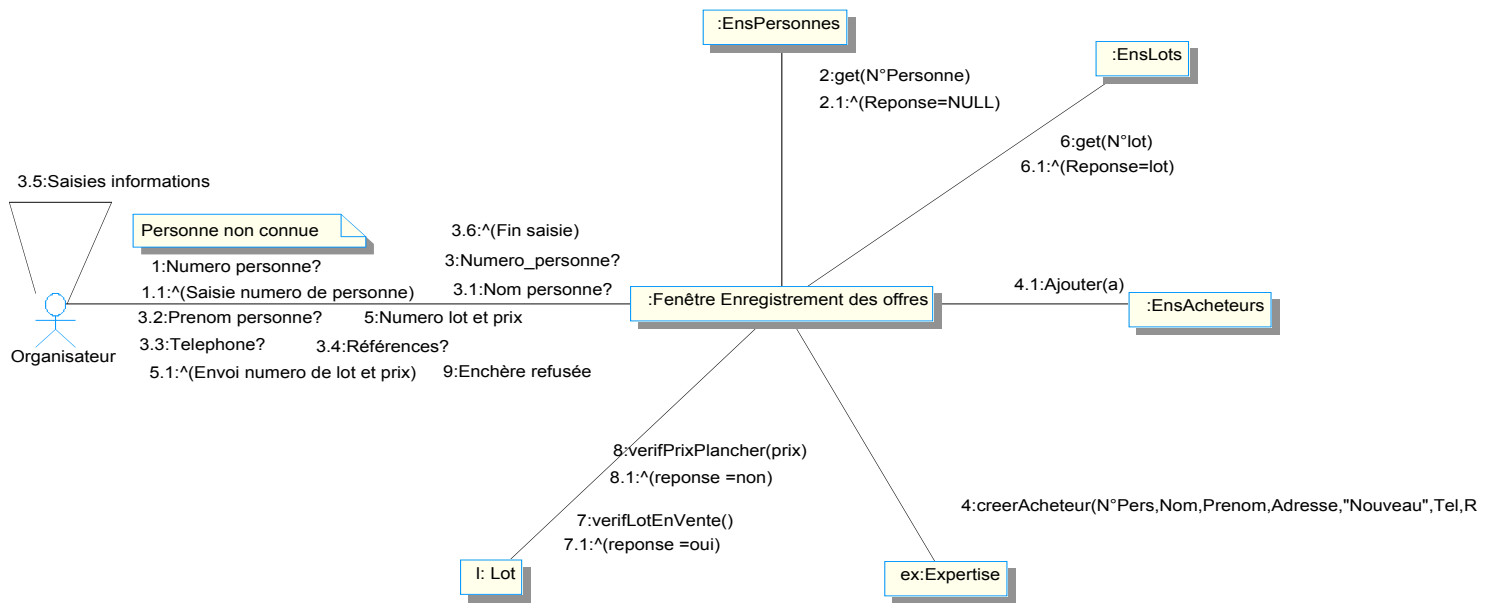
- Personne inconnue et prix proposé < au prix planché

Enregistrement des enchères : personne inconnue et prix proposé < au prix planché		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 28/10/2006	Modifié le : 4/11/2006



Enregistrement des enchères : personne inconnue et prix proposé < au prix plancher		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 21/10/2006	Modifié le : 7/11/2006





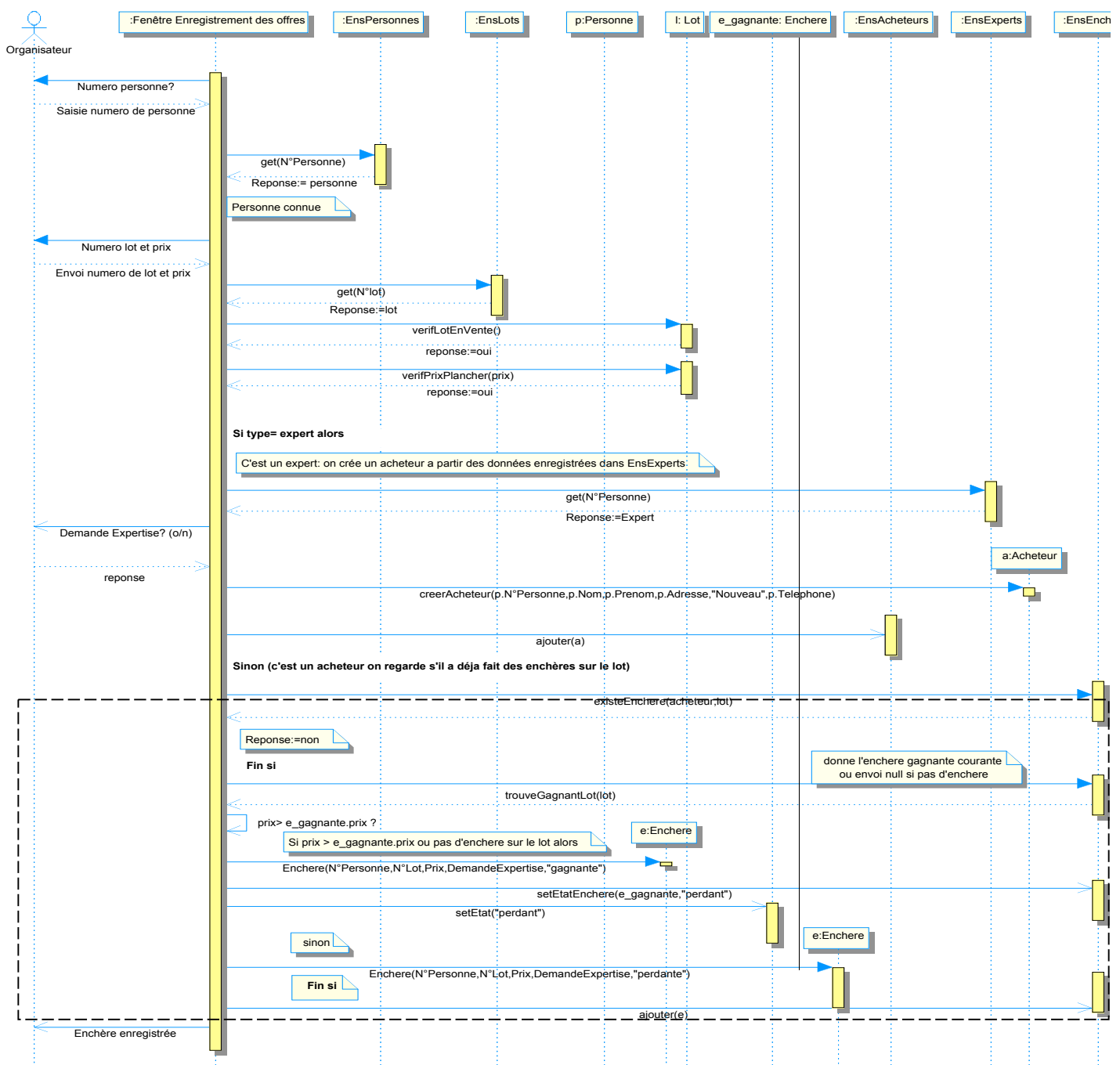


## b) Clôture des ventes

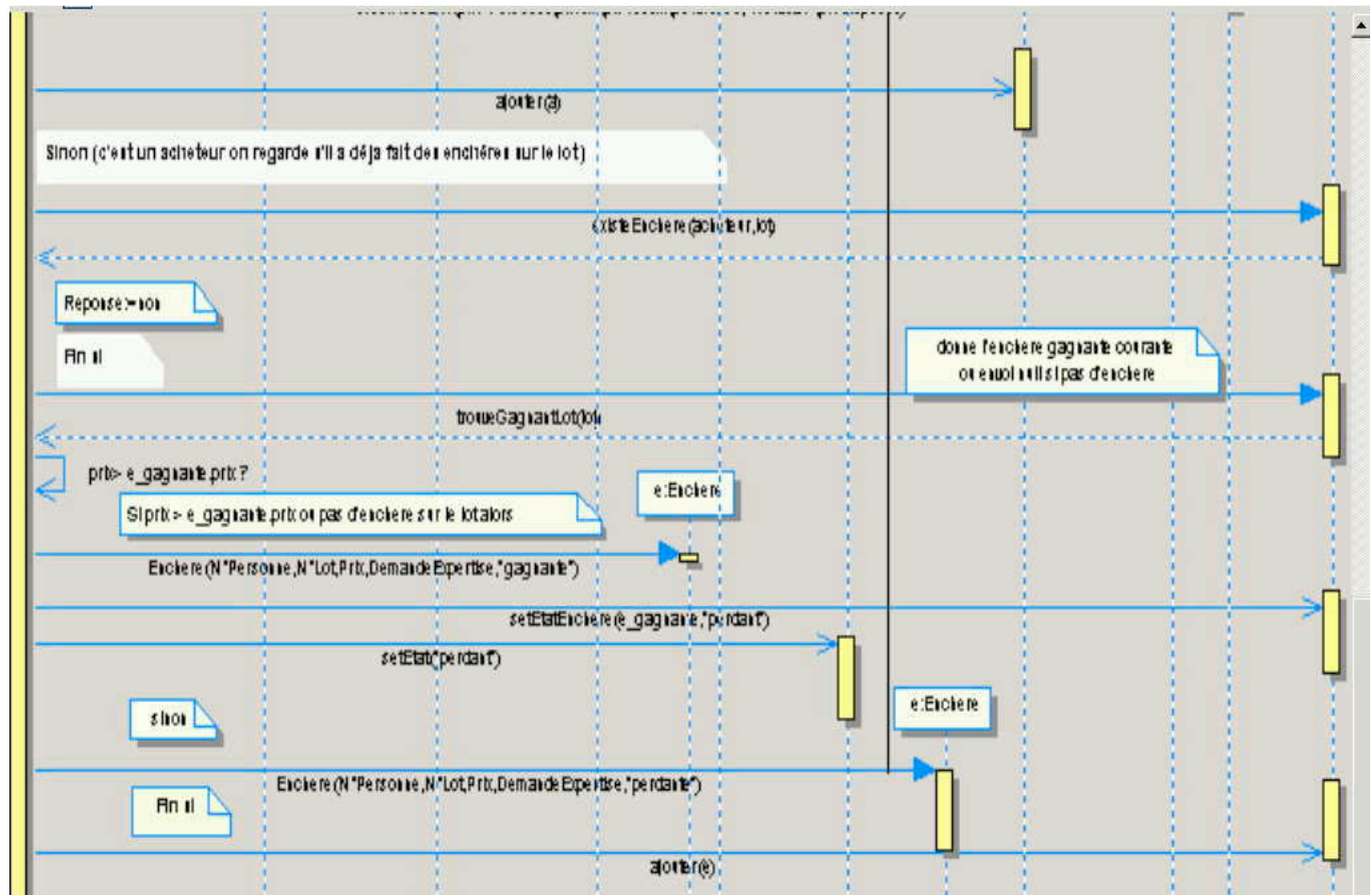
Lors de la clôture des ventes, nous nous sommes rendu compte qu'à aucun moment nous ne mettons l'enchère à l'état « gagnant ». Nous avons donc revu la procédure d'enregistrement des offres et avons décidé que lorsque la création est possible, nous allons comparer la nouvelle enchère à la courante enchère gagnante et si elle est plus intéressante ou si c'est la première enchère faite sur le lot, on la crée avec un état = « perdante » sinon avec un état = « gagnante ».

Ce qui donne :

Enregistrement des offres: personne connue et type <> vendeur		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 21/10/2006	Modifié le : 4/11/2006

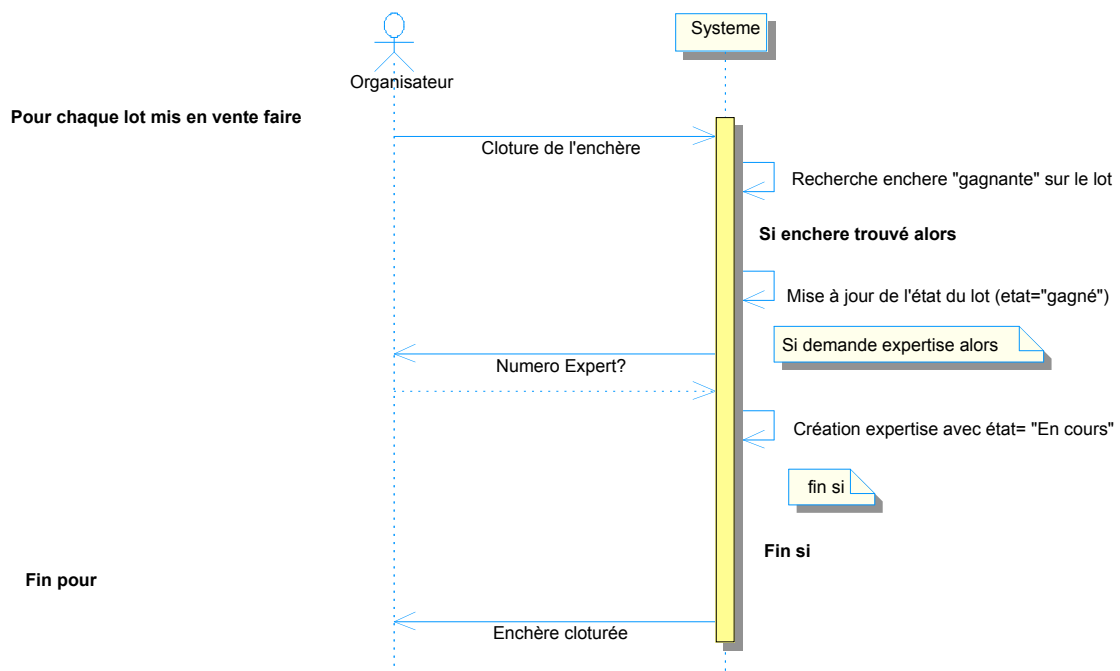


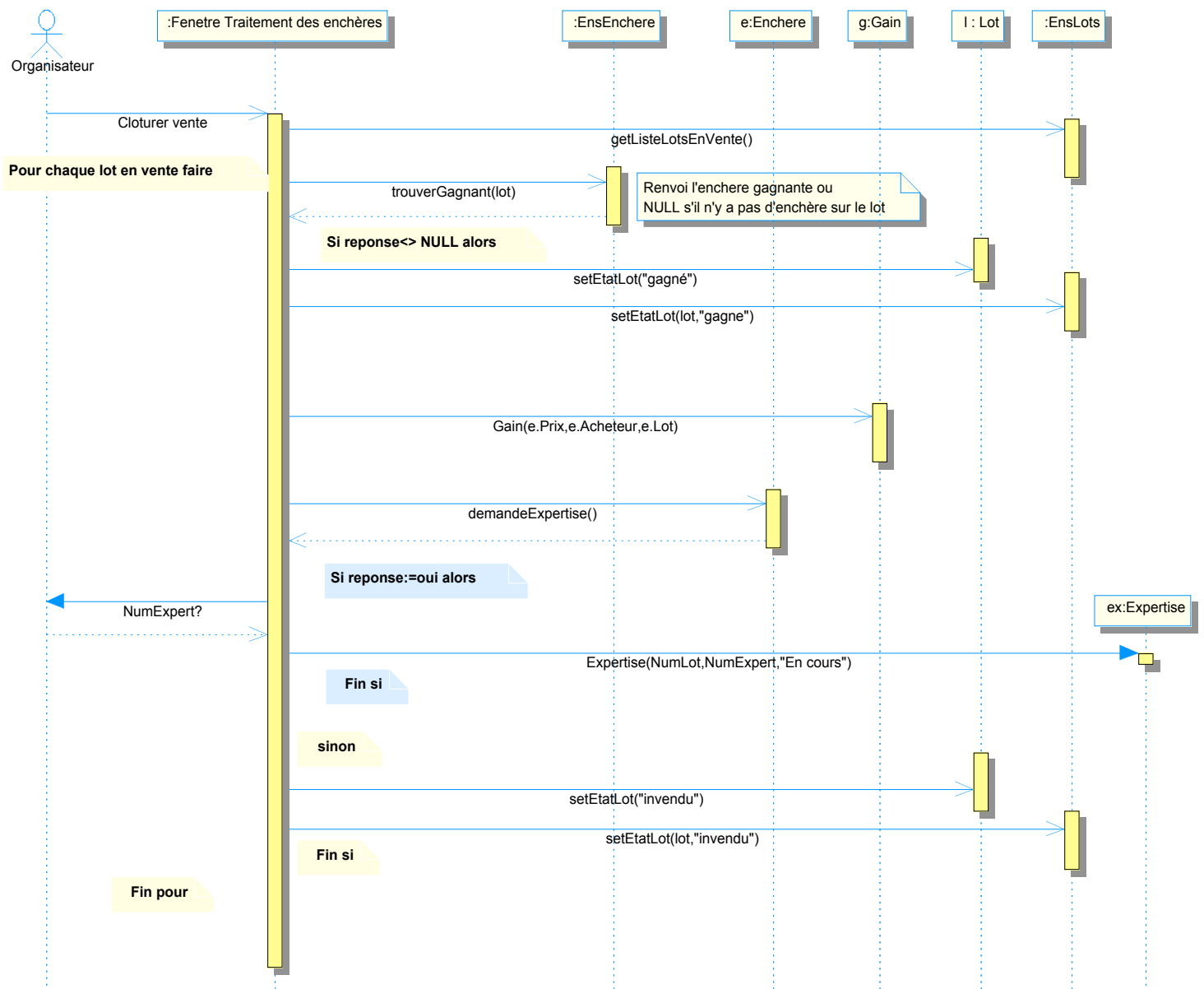
## Partie zoomée



**(1) Diagrammes de scénario**

Cloture des ventes		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 21/10/2006	Modifié le : 4/11/2006



**(2) Diagrammes de séquence****c) Edition des résultats****2. Gestion des expertises**

La gestion des expertises comprend la saisie des expertises effectuées et la consultation des différentes expertises : en cours, nouvelle ou ancienne.

**a) Saisie des expertises**

Cette fonctionnalité a pour objectif de saisir les résultats des expertises demandées sur les lots remontés.

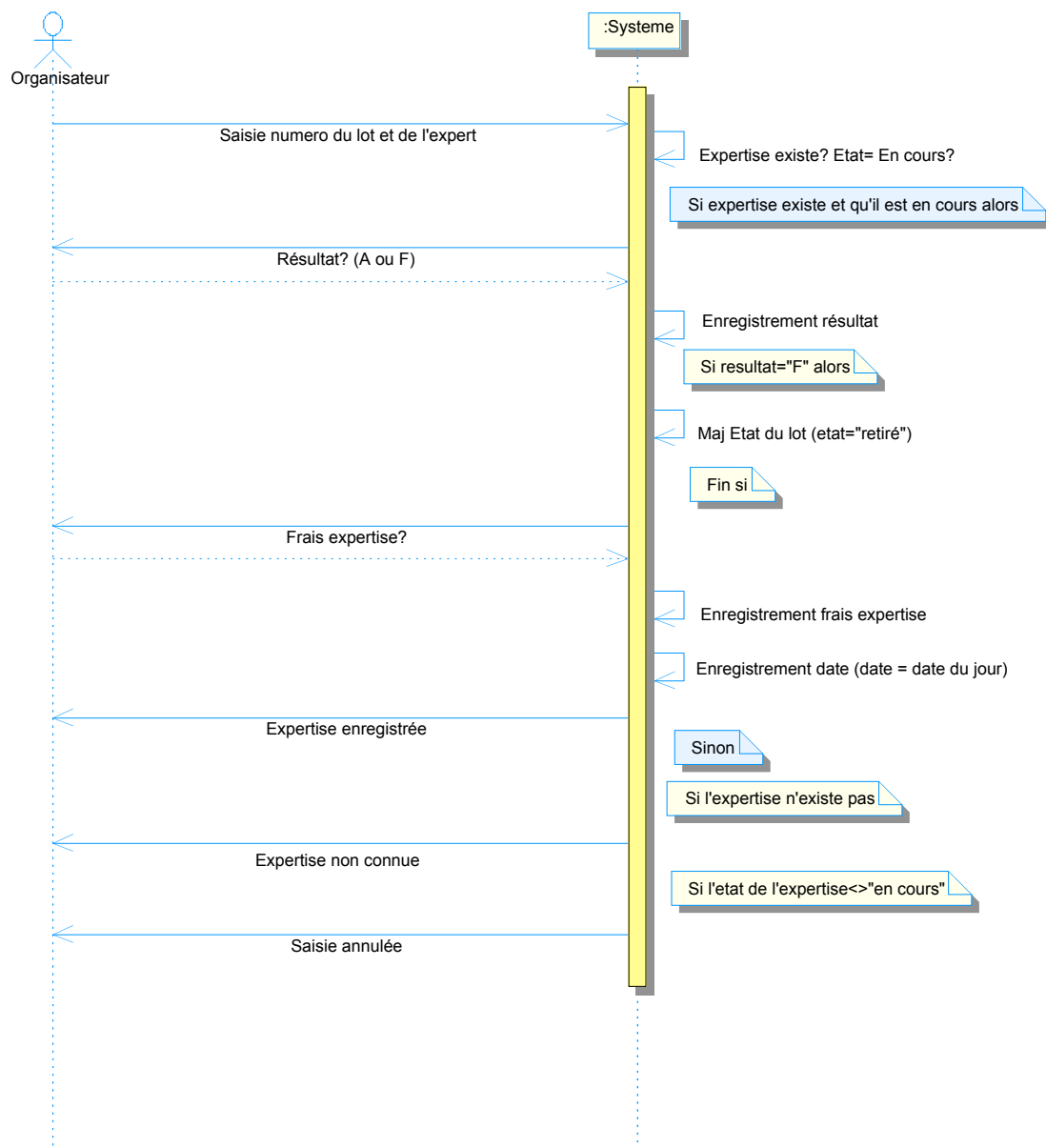
**Description du scénario**

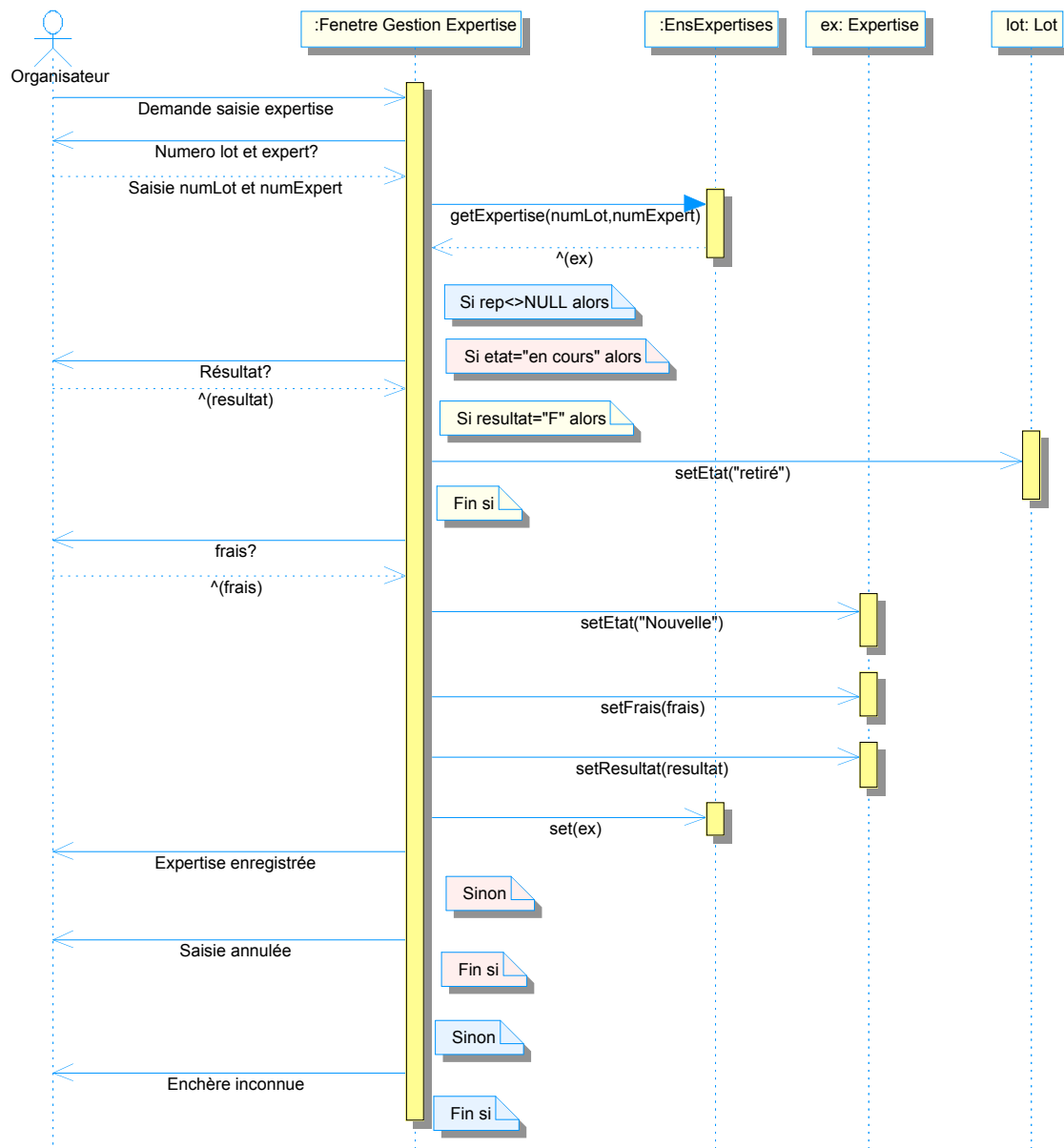
- Récupération du numéro du lot expertisé ainsi que le numéro d'expert.
- Vérification de l'existence de l'expertise et de son état.
  - Si elle n'existe pas ou si l'état  $\neq$  en cours, le traitement est interrompu.

- Sinon on demande la saisie du résultat de l'expertise, « A » ou « F » symbolisant respectivement que le lot est authentique ou faux. et des frais d'expertise
- L'expertise est enregistrée avec une date correspondant à la date du jour et un état = N (nouvelle).
- Si le lot n'est pas authentique, il y a mis à jour de l'état du lot qui passe à « Retiré ».

### (1) Diagramme des scénarios, des séquences et de collaboration

saisie_expertise_scenario		
Projet : UML		
Auteur : I012		
Version : 1	Créé le : 19/10/2006	Modifié le : 4/11/2006





### 3. Gestion des règlements d'achat

#### a) Facturation

Cette fonctionnalité permet la préparation à l'expédition des lots achetés.

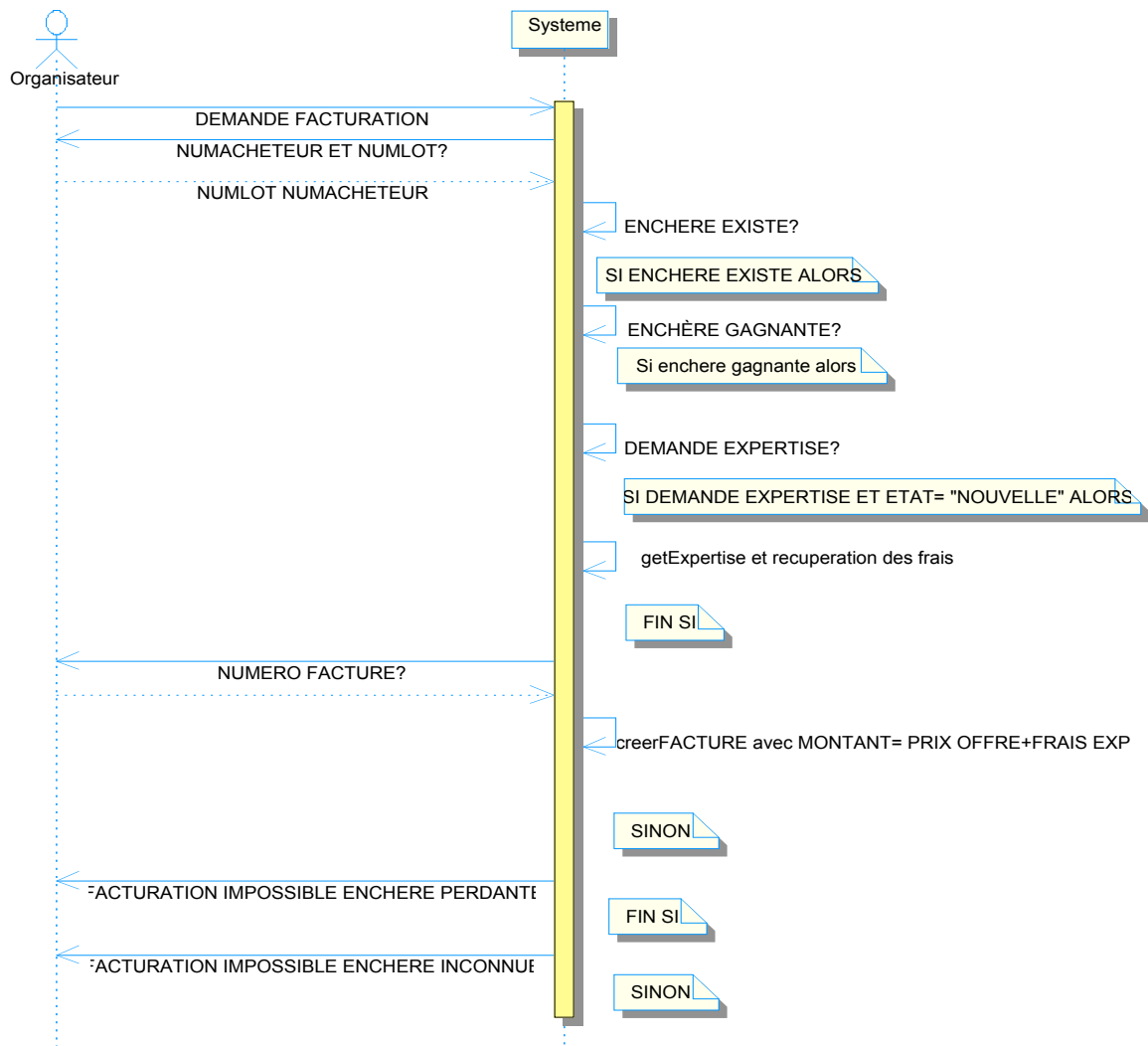
##### (1) Diagrammes de scénario

Pour chaque lot gagné :

- On récupère l'enchère.
  - S'il y a demande d'expertise alors on récupère les frais d'expertises si l'expertise a été déjà faite (vérification de l'état de l'expertise qui doit être nouvelle).
  - On récupère le numéro de la facture en affichant au préalable les infos sur l'enchère.
  - Et on crée la facture.

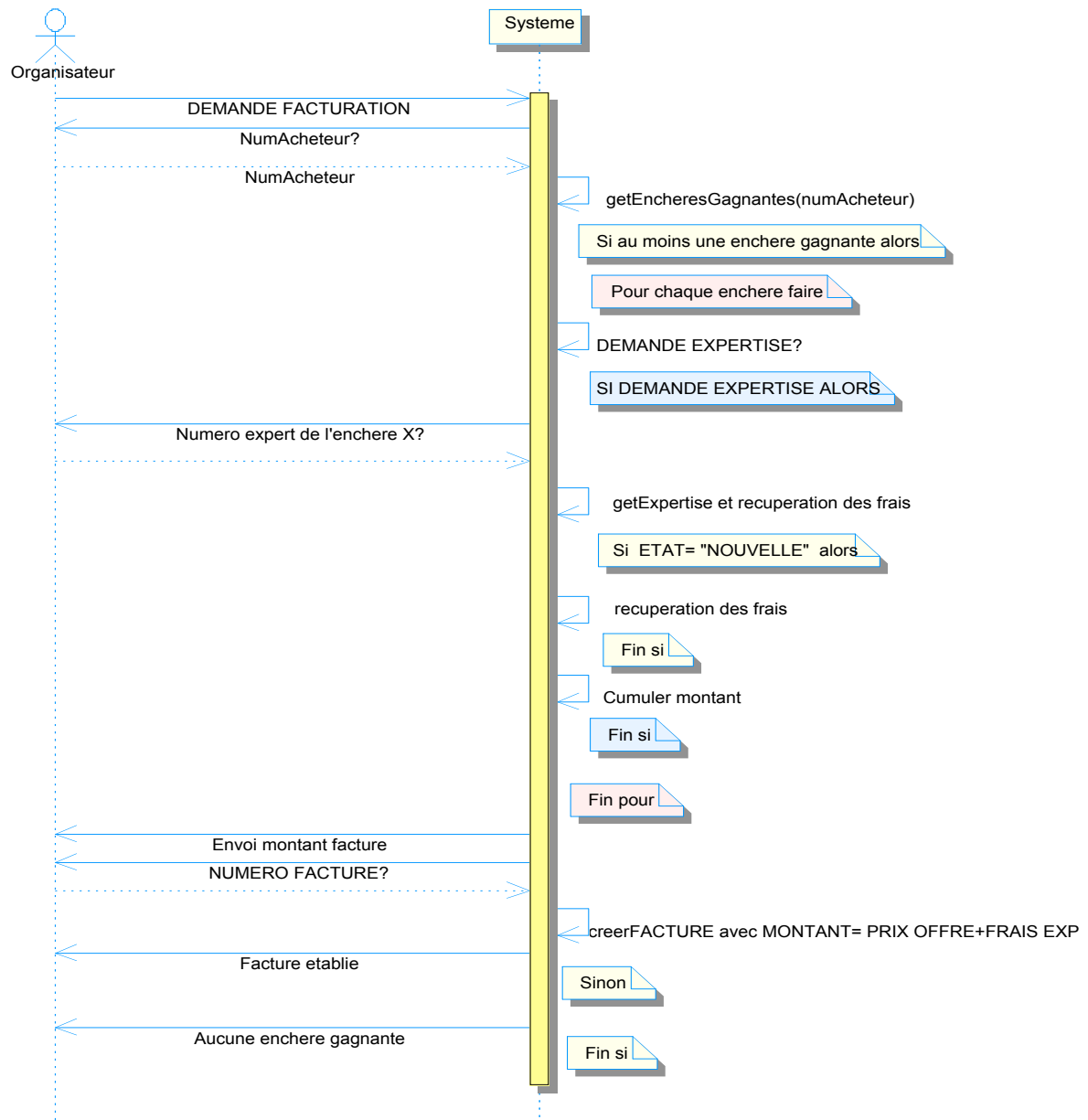
Pour le premier diagramme, nous avons voulu effectuer les factures pour un acheteur à chaque fois et ceci pour un lot.

facturation_scenario		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 4/11/2006	Modifié le : 4/11/2006



Cependant, en réalité il n'y a pas une facture par lot, il y a une facture par acheteur et après chaque clôture de vente s'il a au moins une enchère gagnante. Nous avons donc modifié le scénario en cumulant les montants de chaque enchère gagnante de l'acheteur.

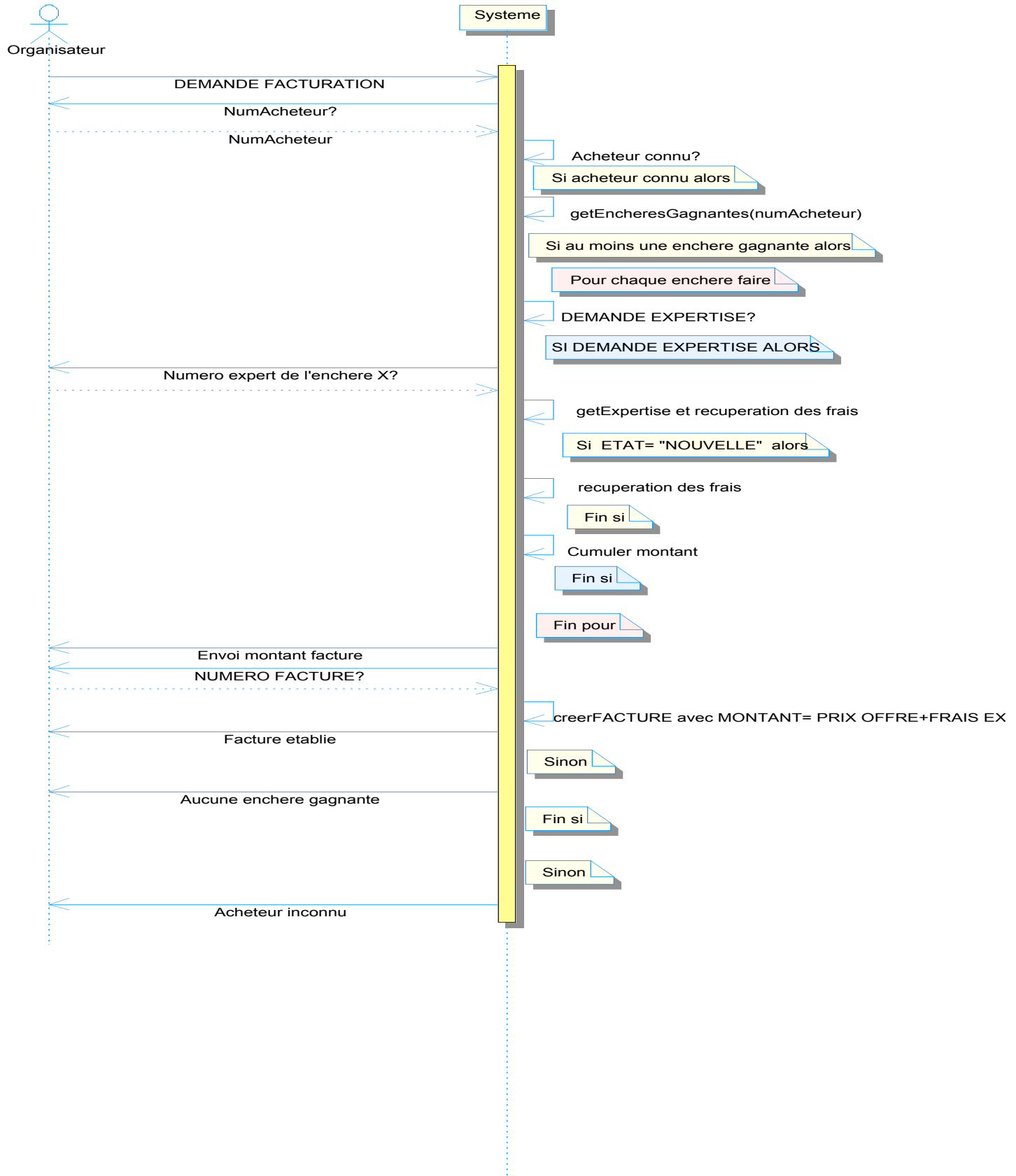
facturation_scenario		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 4/11/2006	Modifié le : 4/11/2006



Cependant, nous n'avons pas géré le cas où l'acheteur saisi par l'organisateur n'ait pas d'enchère gagnante et le cas où le lot saisi n'existe pas.

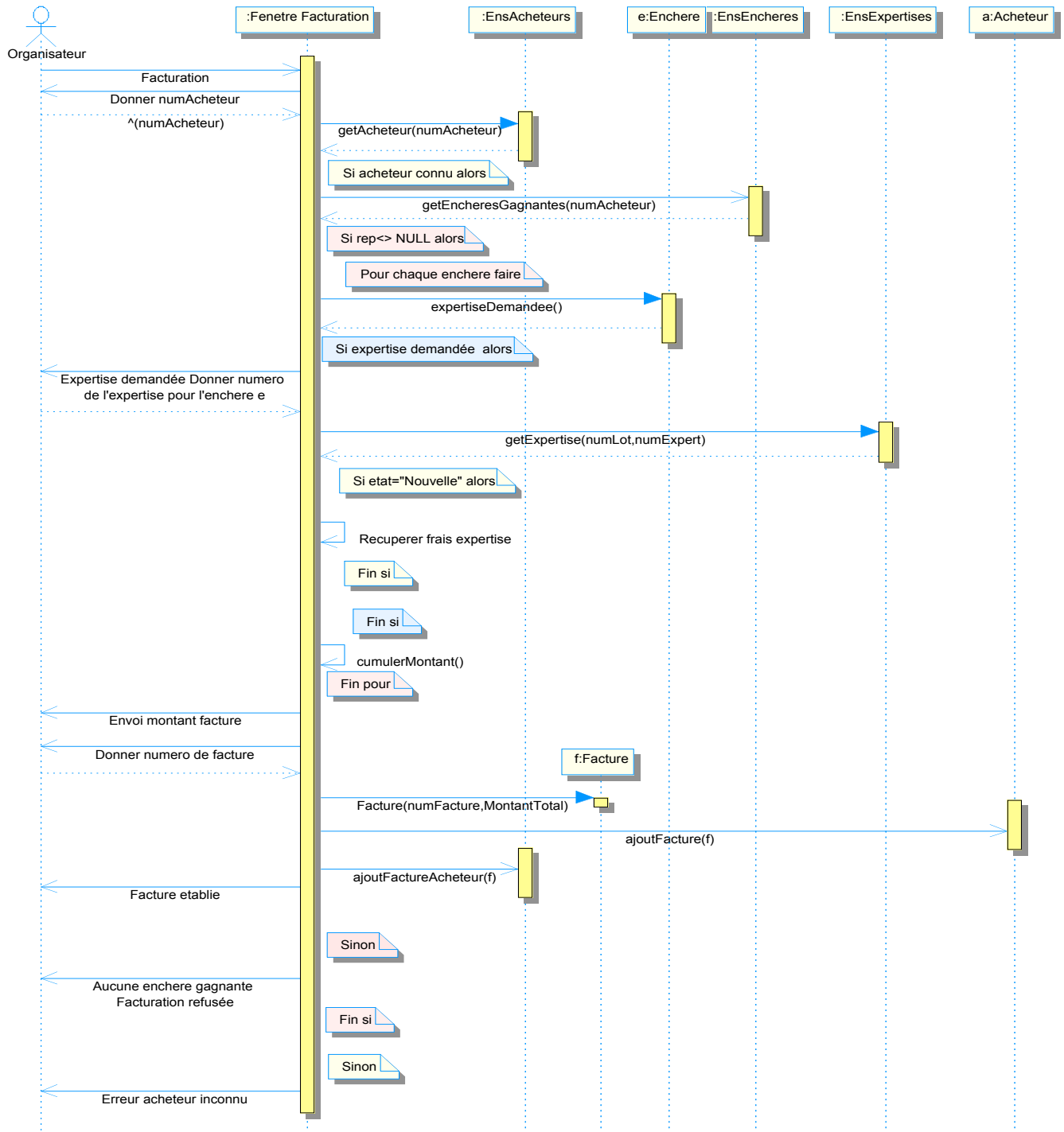


facturation_scenario		
Projet : UML		
Auteur : FAGBEGNON		
Version : 1	Créé le : 4/11/2006	Modifié le : 7/11/2006



**(2) Diagrammes de séquence**

Les classes utiles pour ce traitement sont : la fenêtre Facturation, Enchere, Lot, Acheteur et Facture.



## **b) Encaissement**

C'est l'enregistrement des différents règlements des achats.

### ***(1) Diagrammes de scénario***

Le scénario pour ce traitement comprend :

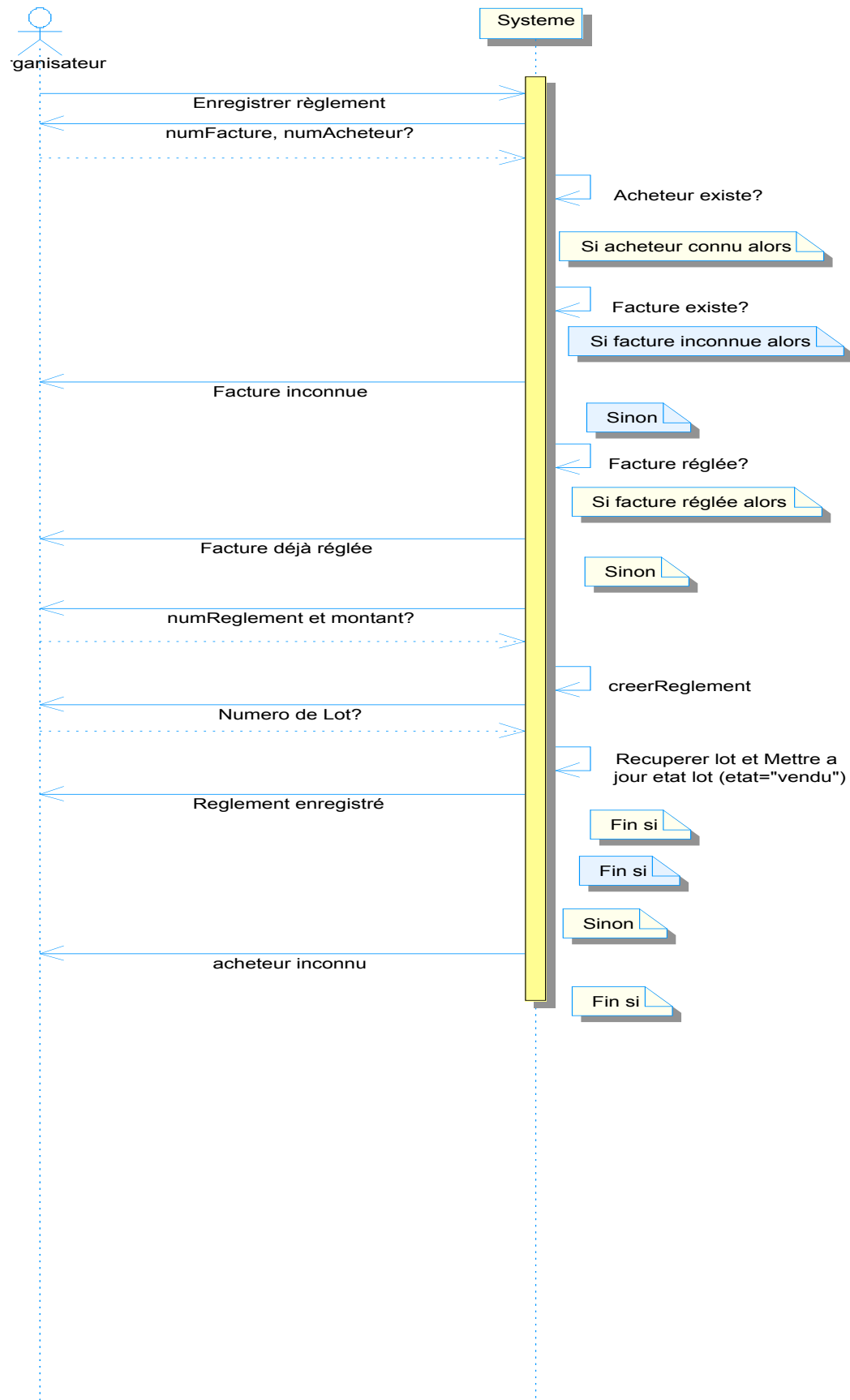
- Une étape de saisie du montant du règlement ainsi que son numéro.
- La création du règlement.
- La mise à jour de l'état du lot qui sera désormais vendu.

Cette méthode ne permet pas d'assurer la cohérence des données ; il peut avoir des règlements sans facture et le règlement ne correspondra à aucun acheteur.

Il faut donc :

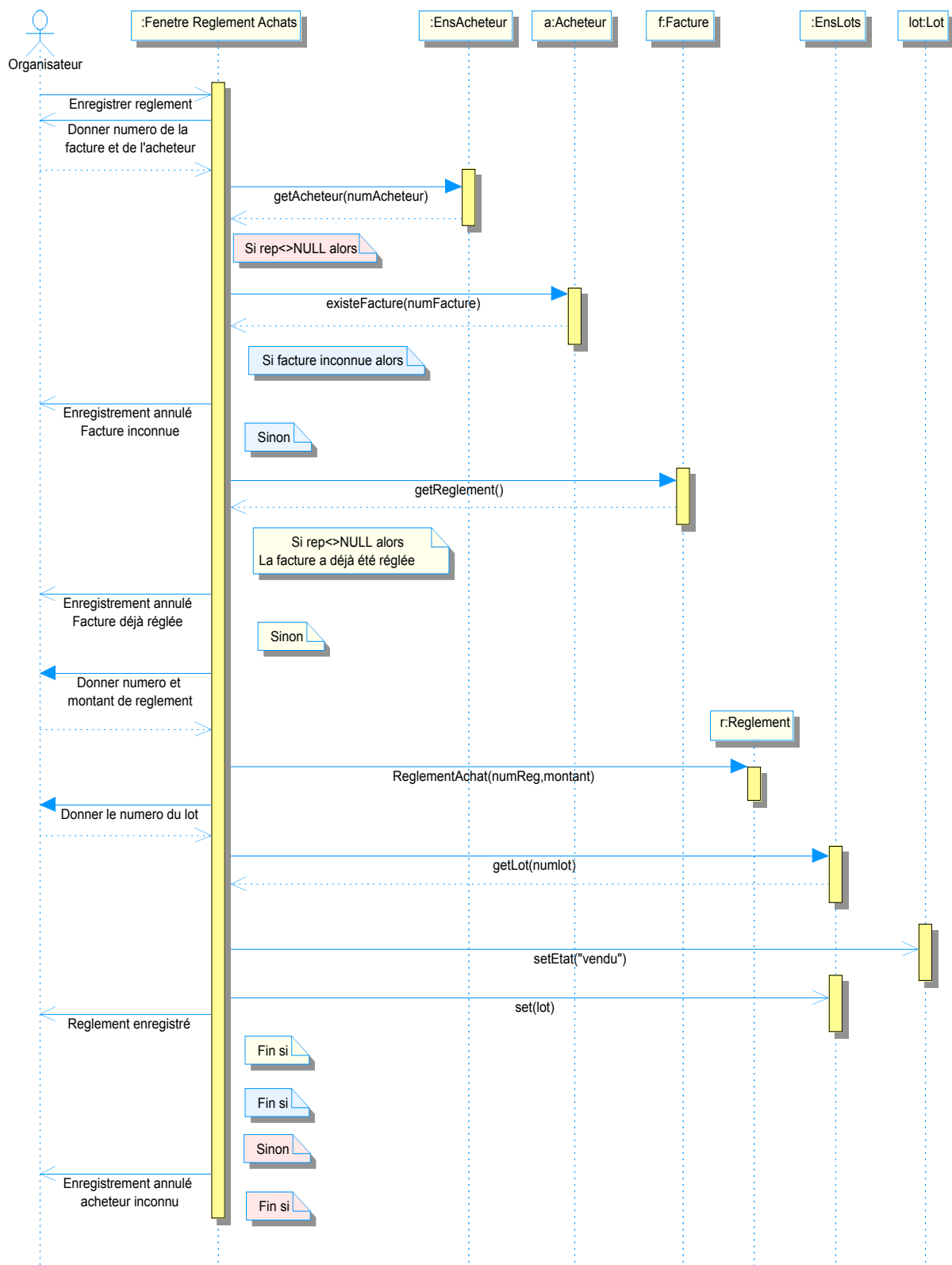
- Vérifier s'il y a une facture correspondant au règlement. Pour cela, nous demandons à l'organisateur le numéro de la facture correspondant au règlement ; ensuite nous vérifions si elle existe. Si c'est le cas on procède à la saisie du règlement sinon on arrête le traitement.
- Récupérer le numéro de l'acheteur, vérifier et s'assurer qu'il est connu.

C'est une fois ces conditions remplies que le règlement est créé.



**(2) Diagrammes de séquence**

Le traitement se fait par le biais de la fenêtre Encaissement, et les classes Acheteur, Facture et Lot doivent être renseignés. Il y aura création d'un objet « ReglementAchat ».



## 4. Payement des vendeurs

### a) Règlement

## 5. Mise à jour des Personnes et des Lots

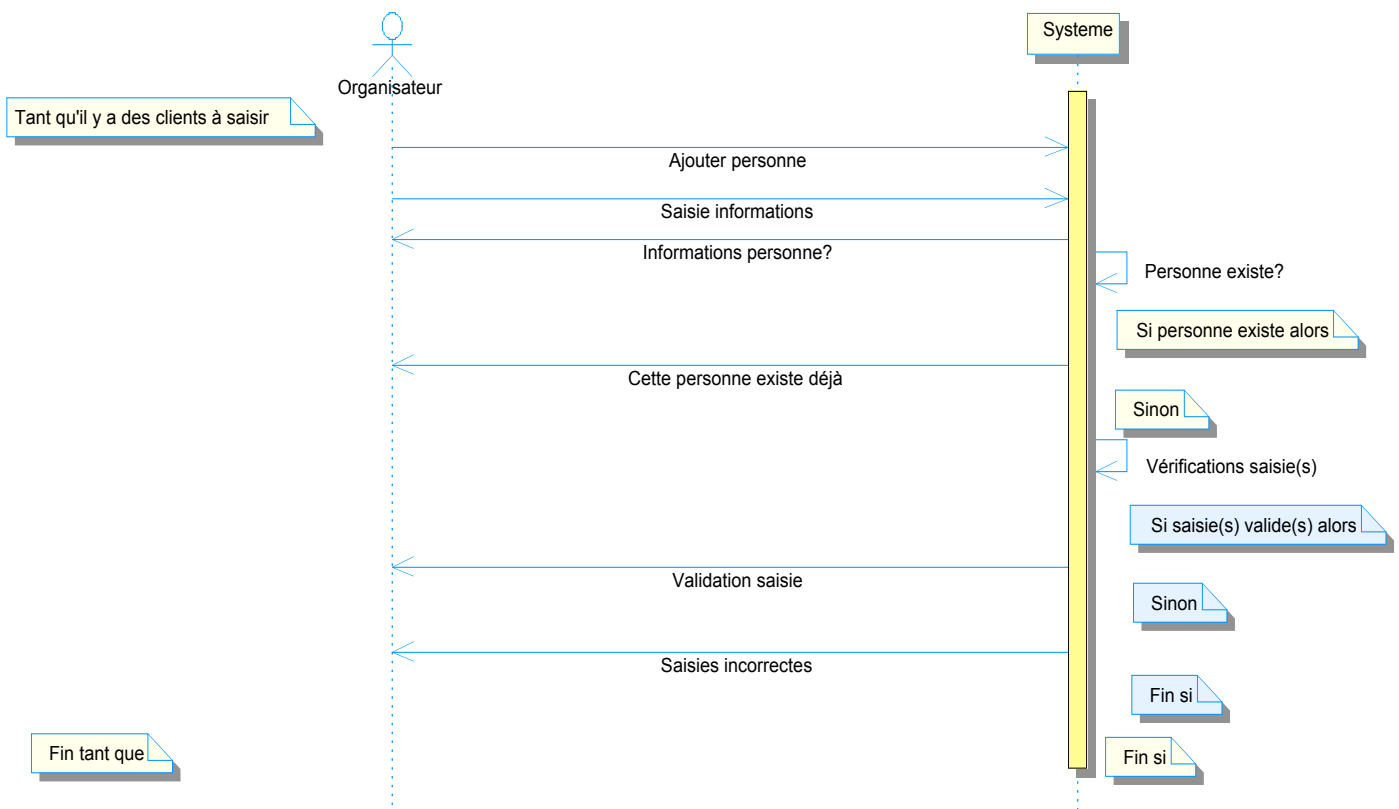
La mise à jour des personnes et des lots suivent le même principe ou plutôt ont les mêmes contraintes. Pour illustration nous allons nous limiter à l'explication des scénarios d'ajout, de suppression et de modification d'une « Personne » en général. Les informations saisies changeront selon que la personne est expert, vendeur ou acheteur ; elles seront aussi différentes s'il s'agit d'un lot.

Le diagramme de séquences n'apporte rien de nouveau si ce n'est la spécification des classes et des méthodes utilisées pour les différentes fonctions.

### a) Ajout

L'ajout d'un lot ou d'une personne est possible si le lot ou la personne n'existe pas déjà et si les saisies sont valides.

<b>ajout des clients</b>
Projet : UML



Classes utiles pour le diagramme de séquence :

- Personne, Expert, EnsExperts et EnsPersonnes si c'est un expert.

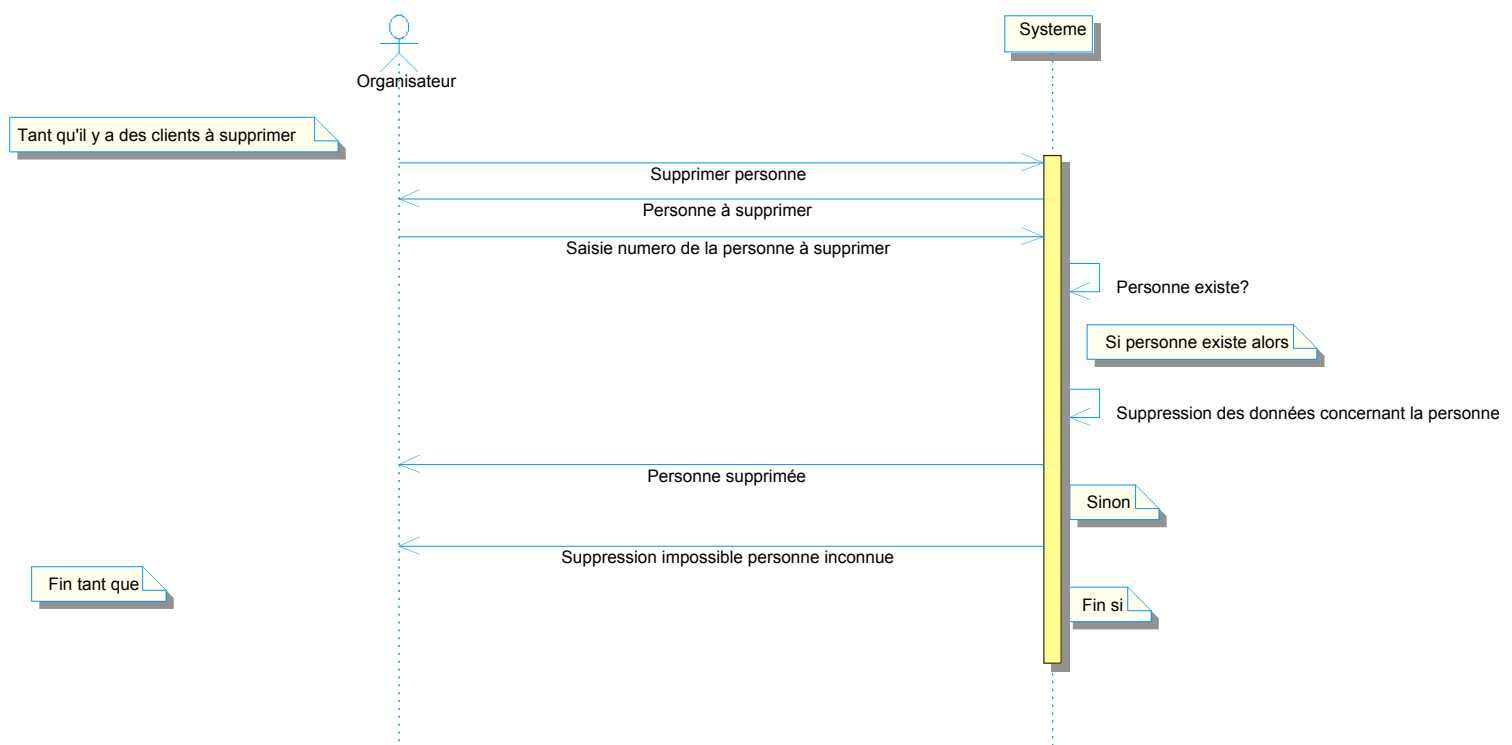
- Personne, Acheteur, EnsAcheteurs et EnsPersonnes si c'est un acheteur.
- Personne, Vendeur, EnsVendeurs et EnsPersonnes si c'est un vendeur.
- Lot et EnsLots si c'est un lot.
- Les classes Interfaces.

En effet, il faut créer les objets et les ajouter dans les classes « Ensemble ». Ces dernières serviront également à vérifier l'existence des éléments à ajouter.

## b) Suppression

Pour qu'un élément soit supprimé, il faut qu'il existe.

<b>suppression personne</b>
Auteur : I012



Classes utiles pour le diagramme de séquence :

- Personne, Expert, EnsExperts et EnsPersonnes si c'est un expert.
- Personne, Acheteur, EnsAcheteurs et EnsPersonnes si c'est un acheteur.
- Personne, Vendeur, EnsVendeurs et EnsPersonnes si c'est un vendeur.
- Lot et EnsLots si c'est un lot.
- Les classes Interfaces.

Il faut supprimer les objets et les supprimer dans les classes « Ensemble ». Ces dernières serviront également à vérifier l'existence des éléments à supprimer.

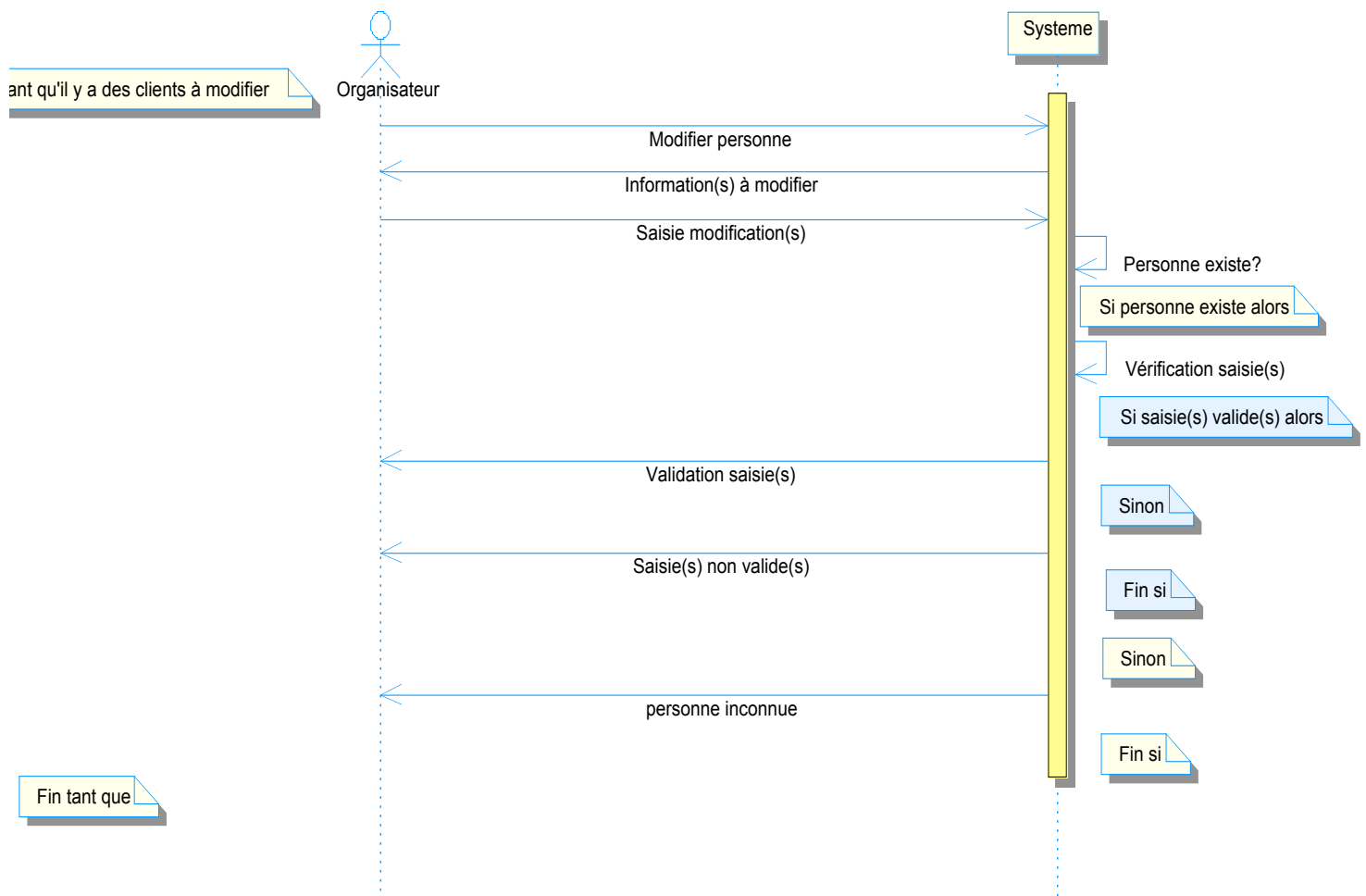
La suppression des infos les concernant va dépendre des choix de Phildex. Pour l'expert par exemple, Phildex peut choisir de garder toutes les expertises d'un expert même s'il n'a plus de contrat avec lui. Dans ce cas, la suppression n'est pas indispensable, dans le cas contraire il faudra la faire pour garder la cohérence entre les données. Cependant la première hypothèse semble plus réaliste.

On a pas besoin de supprimer physiquement les infos les concernant car il s'agira de modifier l'état des objets. Si c'est une Personne l'état passera « parti » et si c'est un lot l'état passera à « retiré ». Nous avons fait le choix d'effectuer une suppression « logique » des personnes et des lots.

### c) Modification

Pour qu'un élément soit modifié, il faut qu'il existe.

<b>Modification personne</b>
Projet : UML



Classes utiles pour le diagramme de séquence :

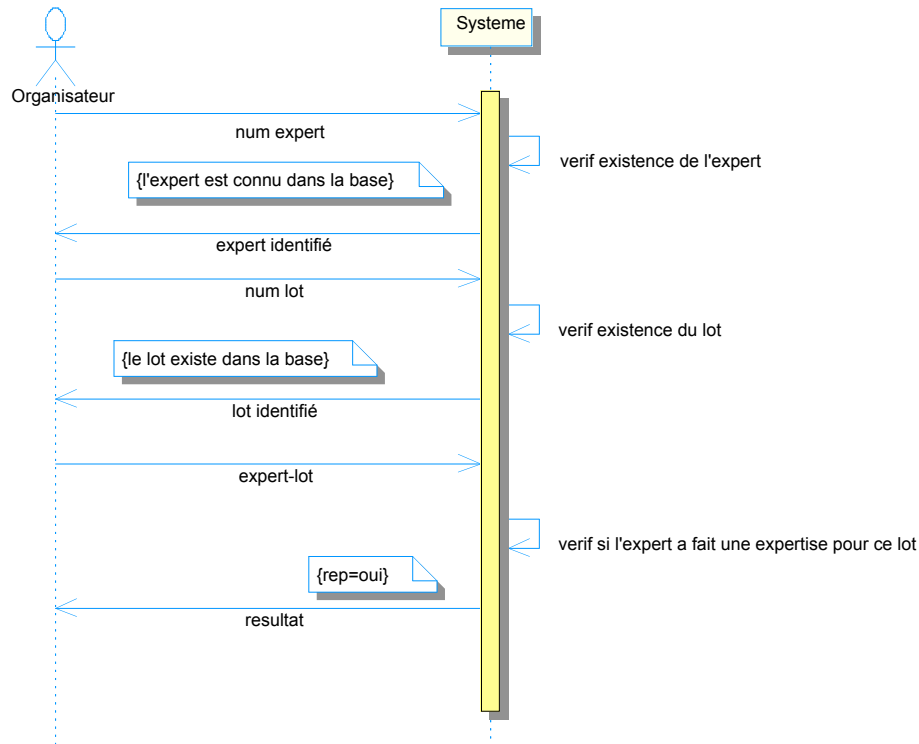
- Personne, Expert, EnsExperts et EnsPersonnes si c'est un expert.
- Personne, Acheteur, EnsAcheteurs et EnsPersonnes si c'est un acheteur.



- Personne, Vendeur, EnsVendeurs et EnsPersonnes si c'est un vendeur.
- Lot et EnsLots si c'est un lot.
- Les classes Interfaces.

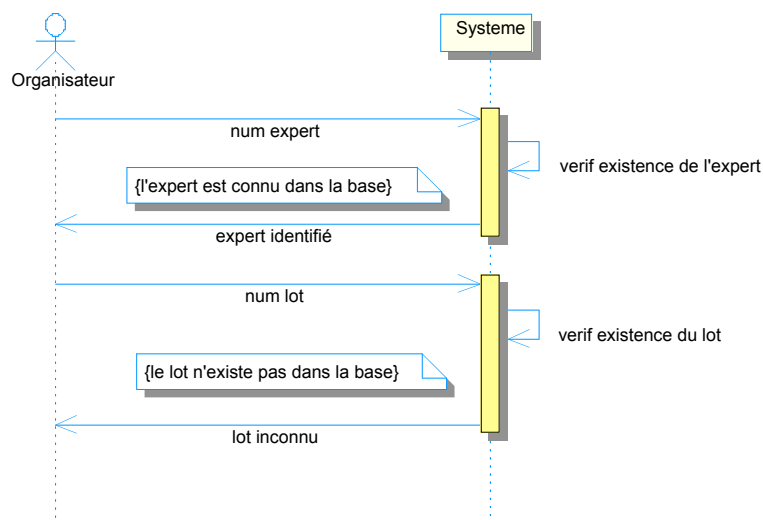
#### d) Consultation des résultats

- Cas nominal

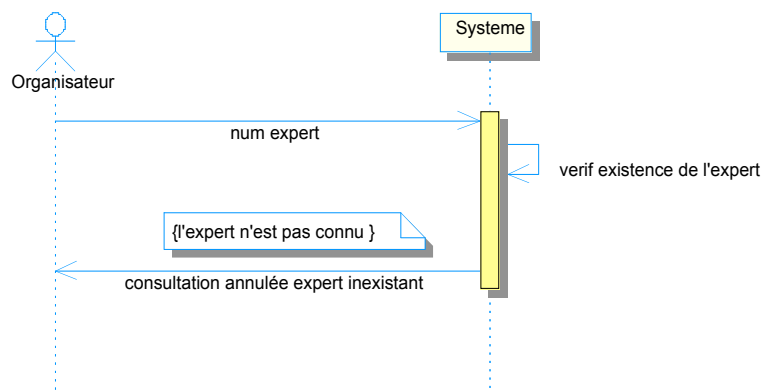


- Cas lot inconnu

Si l'expertise n'existe pas il y a message d'erreur.

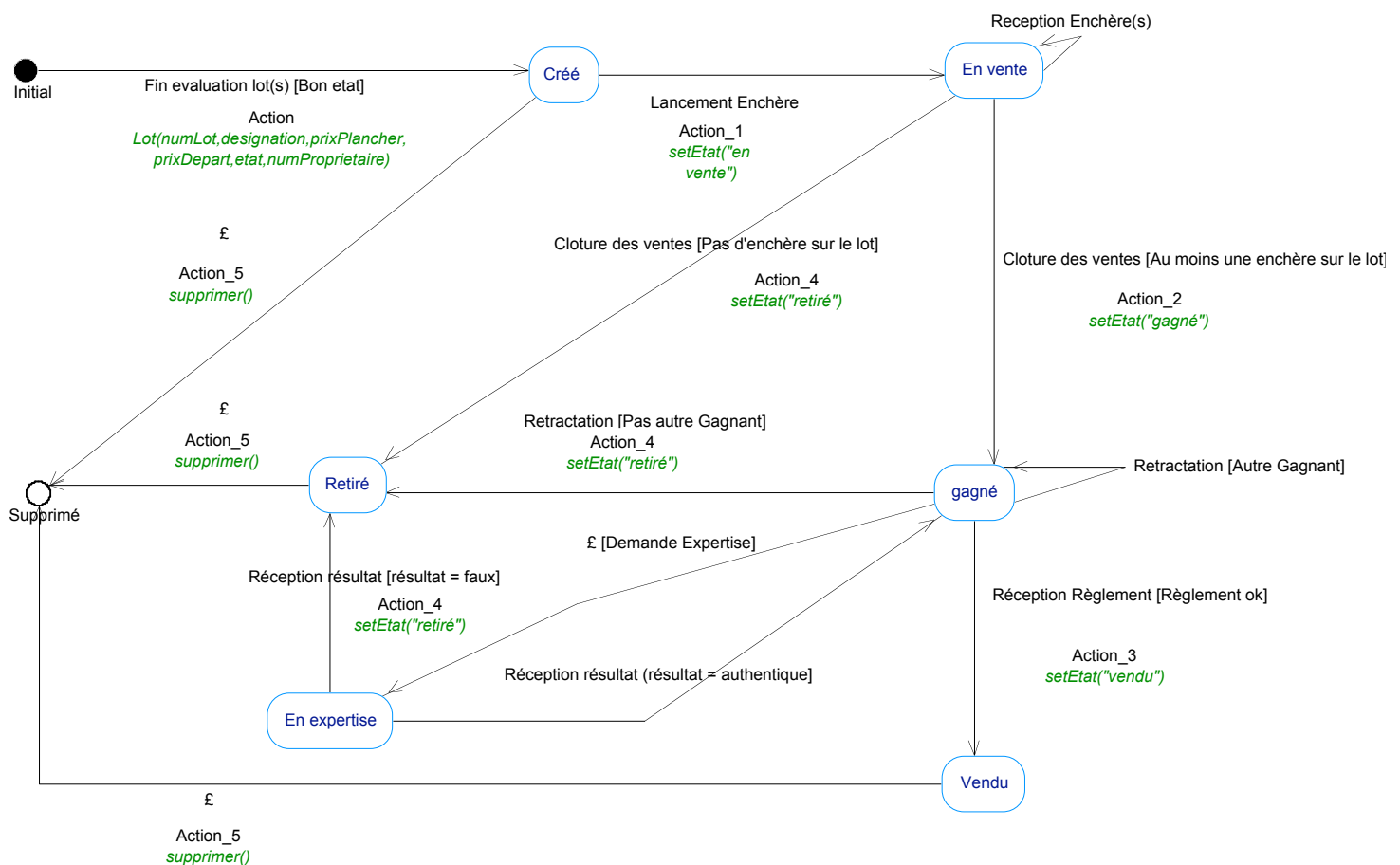


- Cas expert inconnu



## D. Diagramme états transition

### 1. Lot



Selon notre diagramme de classes, un lot peut être soit « proposé », soit « en vente », soit « retiré », soit « gagné », soit « vendu ». On retrouve tous ces états dans l'automate ci-dessous cependant, lors de la modélisation il a fallu en rajouter pour vraiment modéliser le comportement d'un lot dans la procédure de vente de Phildex.

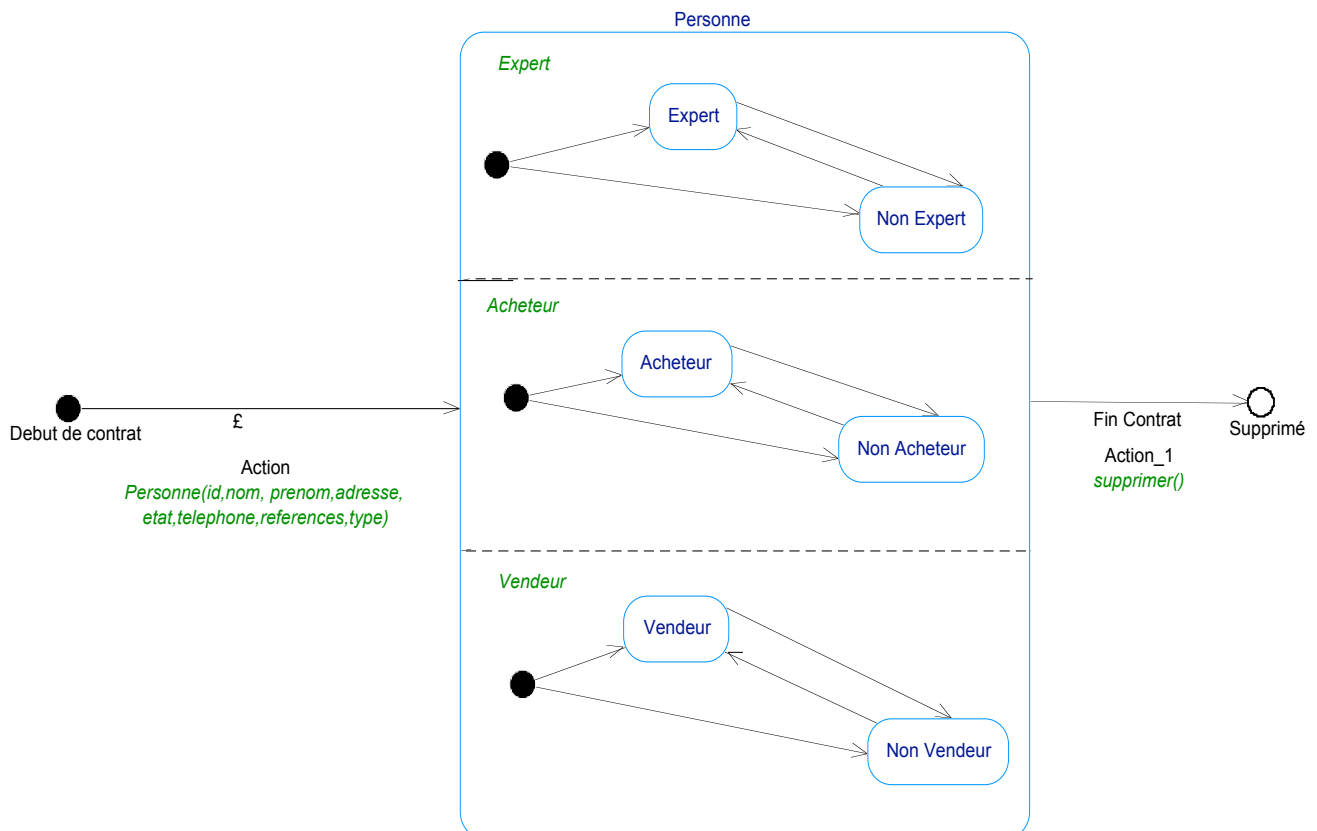
L'état « En Expertise » représente l'état d'un lot pour lequel une expertise a été demandée.

Les ventes sont clôturées au bout d'un mois d'enchères, j'aurais donc pu remplacer l'événement « clôture des ventes » par l'écoulement d'une période d'un mois.

Les lots qui font l'objet d'une expertise peuvent être retirés lorsque le résultat de l'expertise stipule que l'objet mis en vente n'est pas authentique.

## 2. Personne

Une personne peut être Vendeur, Acheteur, Expert, Vendeur Acheteur, Expert Acheteur, Vendeur Expert, Vendeur Acheteur Expert. Une personne possède donc un état selon le(s) rôle(s) qu'il joue sans la vente. D'où la modélisation par un ensemble d'automates vivants en parallèle, représenté ci-après.

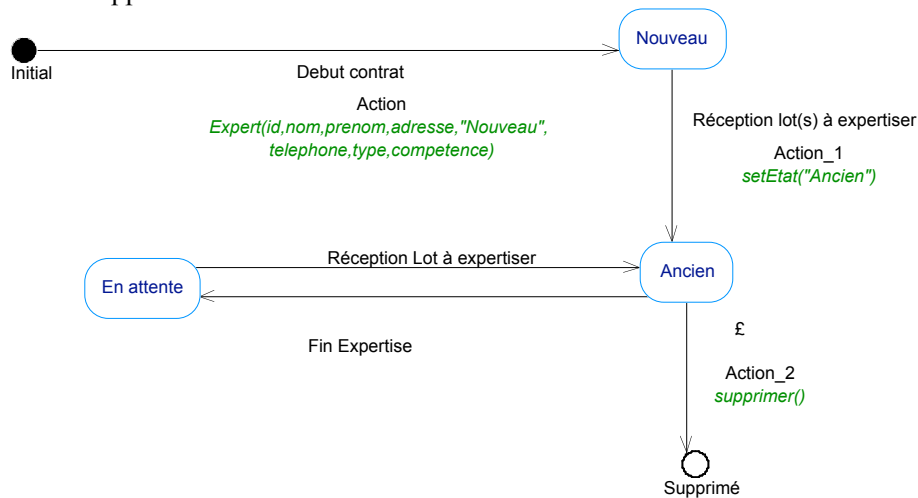


Note : La fin d'un contrat peut être dû à un décès, un départ à la retraite ou une rupture du contrat par l'une des parties.

Nous souhaitons préciser que cette formalisation est plus orientée « réalité » qu'implémentation. En effet, il n'y aura jamais un objet « Personne » qui respecte ce parallélisme, seulement il peut avoir des objets « Acheteur » qui ont le même numéro que objets « Vendeur » ou/et qui ont le même numéro que des objets « Expert ».

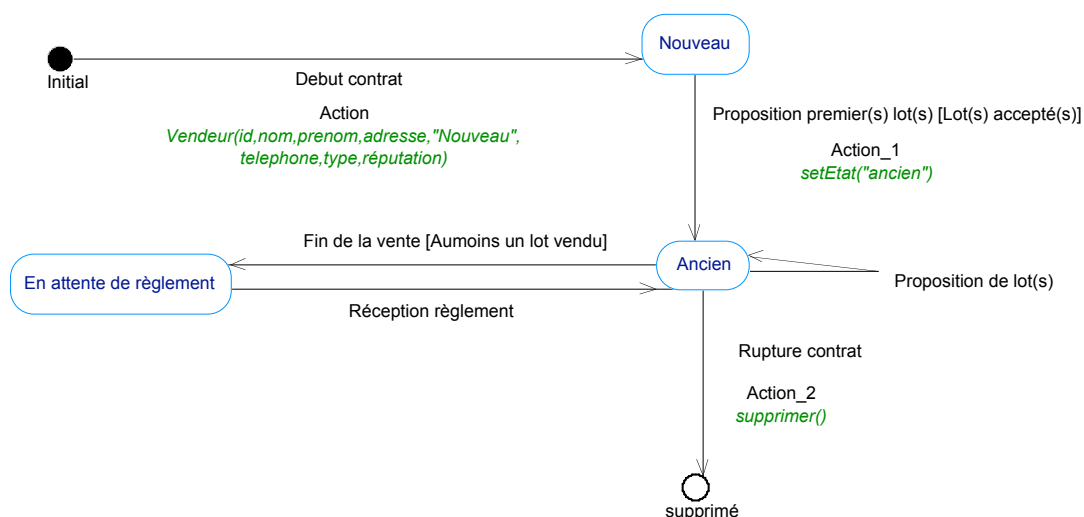
### 3. Expert

Un expert est soit « nouveau » soit « ancien ». L'état « Ancien » correspond à un expert qui a déjà effectué une ou des expertise(s) pour le magasin et l'état « Nouveau » correspond à un expert qui vient de signer un contrat avec Phildex et qui n'a pas encore fait d'expertise. J'ai rajouté un état « En attente » pour symboliser un expert qui finit ses expertises et qui attend une attribution. Le contrat peut être rompu soit par l'une des parties soit pour cause de décès ou de retraite ; l'expert est alors à l'état « Supprimé ».



### 4. Vendeur

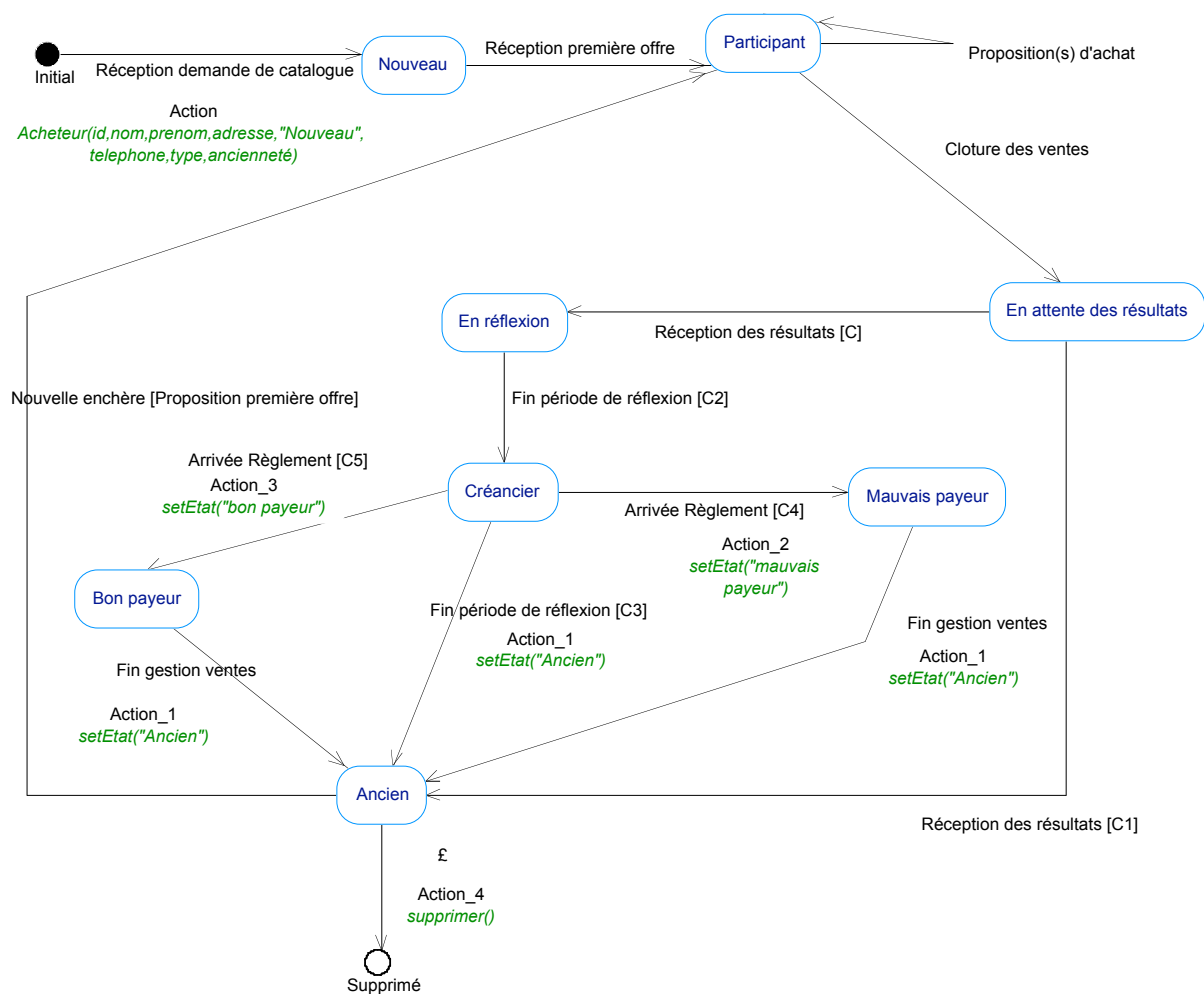
Un vendeur est « Nouveau » lors de son adhésion à Phildex, il devient ensuite « Ancien » lorsqu'il participe à une vente. Il peut faire plusieurs propositions et quitter ensuite à la rupture du contrat qui peut être dû à un décès ou à la demande d'une des parties ; le vendeur est alors à l'état « Supprimé ». Il reçoit un règlement lorsque ses lots proposés sont vendus.



## 5. Acheteur

Un acheteur est soit «nouveau », soit « ancien », soit « mauvais payeur », soit « bon payeur ». J'ai supposé qu'il devient « nouveau » lorsqu'il fait une demande de catalogue et qu'ensuite il devient « ancien » lors de la fin de la première vente à laquelle il a participé. A l'état « ancien » il peut participer à d'autres enchères. Lorsqu'on attend le règlement d'un client, il est à l'état « Créancier ». L'état « En attente des résultats » est l'état de l'acheteur entre la clôture des ventes et l'édition des résultats. A la réception des résultats l'acheteur, s'il est gagnant, dispose d'une période de réflexion et peut se rétracter. Un client est considéré comme « Participant » à la vente dès lors qu'il reçoit le nouveau catalogue.

Il passe à l'état « Supprimé » lorsque aucune proposition d'offre n'a été enregistrée depuis trois années.



C : offre gagnante

C1 : offre perdante

C2 : pas de rétractation

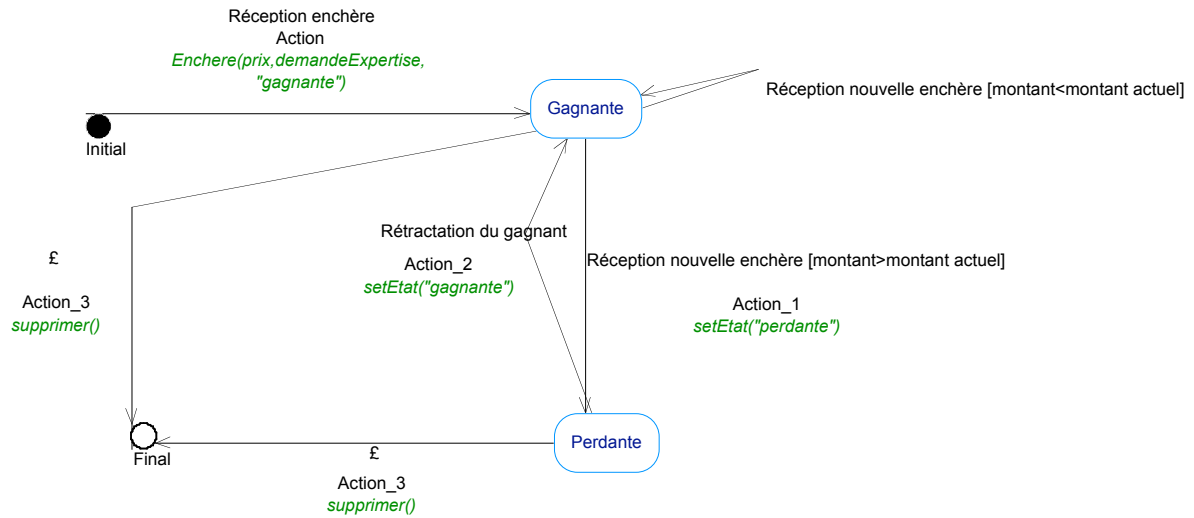
C3 : rétractation

C4: Après délai et/ou règlement pas ok

C5: Dans les délais et/ou règlement ok

## 6. Enchère

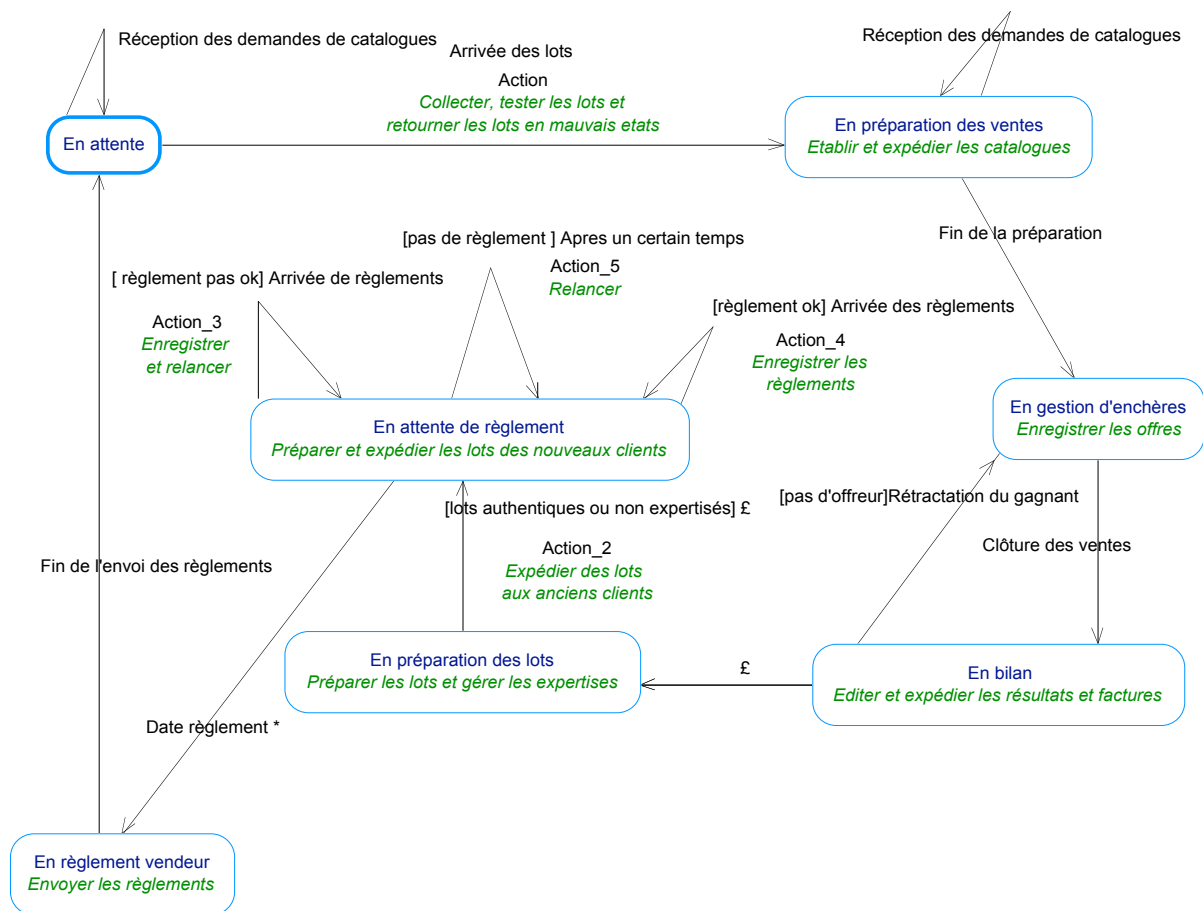
Une enchère ne peut avoir que deux états : « Gagnante » ou « Perdante ». Elle passe d'un état à l'autre soit lors d'une rétractation du gagnant, elle devient dans ce cas perdante, soit lors de la réception d'une offre d'un montant plus élevé, elle devient dans ce cas gagnante. Une offre reste gagnante lorsque aucune nouvelle enchère plus intéressante n'est proposée.



## 7. Organisateur

L'organisateur est un peu comme le coordinateur, le chef d'orchestre. Il collecte les lots, effectue les tests nécessaires et les diverses préparations, gère les enchères, les expéditions des lots et les clients, établit les résultats, la facturation des lots, le paiement des vendeurs et les relances. C'est le « Système ».

Ici nous allons juste proposer un automate qui reflète le fonctionnement du système car faute de temps nous n'avons pas pu faire correspondre les actions aux méthodes.



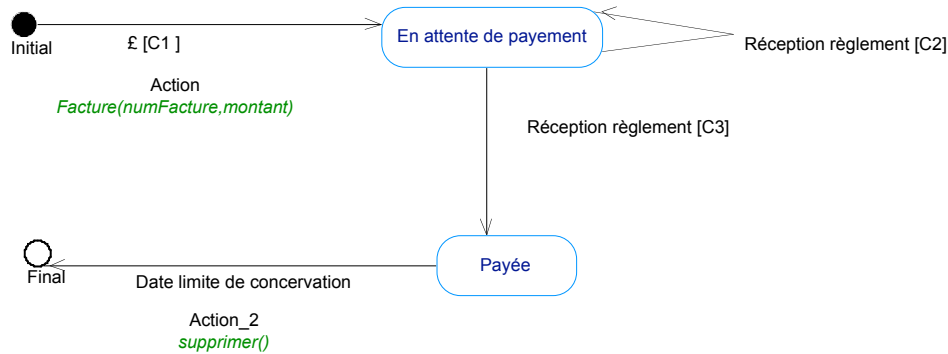
Note : L'état « En attente » est à la fois initial et final.

\* Un mois après la clôture des ventes

J'ai supposé que l'organisateur règle les vendeurs 1 mois après la clôture des ventes même s'il n'a pas encore été totalement réglé par les clients.

## 8. Facture

Une facture est soit en attente de paiement soit payée. Elle est considérée comme payée lorsqu'on reçoit le montant total dû ; donc lorsqu'un règlement lui correspondant est créé. Une facture, n'est créée que pour un ou des lots gagnés et pas en cours d'expertise.



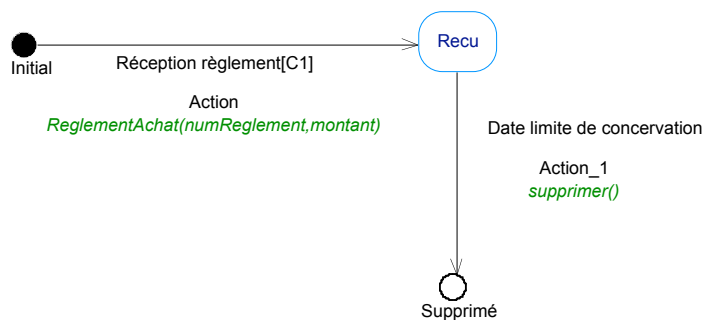
C1 : Enchères sur le ou les lots gagnantes et aucun lot en cours d'expertise

C2 : Montant reçu < montant dû

C3 : Montant reçu = montant dû

## 9. ReglementAchat

Un règlement, a en soit deux états, reçu ou non reçu. L'état non reçu est l'état initial et il est considéré comme « reçu » => création lorsque l'organisateur reçoit le montant total dû et si une facture correspondante existe.

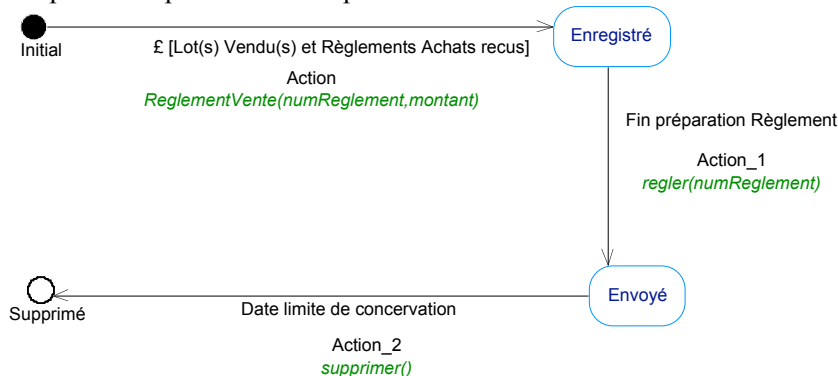


C1 : Montant reçu = Montant dû et la Facture correspondant existe.



## 10. ReglementVente

Nous avons supposé que le règlement des vendeurs se fait seulement si les règlements des clients sont reçus ce qui inclut d'ailleurs que les lots réglés sont vendus. Ceci peut être valable si on considère peut poser des problèmes lorsqu'un contrat



### E. Problèmes rencontrés, évaluation du travail et critiques

La grande difficulté de la notation UML est de garder de la cohérence entre ses différents modèles. Il a donc fallu modifier plusieurs fois les diagrammes lorsque notre vision des choses change ou après considération de certains facteurs non pris en compte.

Nous avons essayé de gérer le maximum de cas, de garder de la cohérence, de respecter les contraintes de gestion spécifier dans le sujet. Cependant, il a fallu faire des choix et des hypothèses sur des points confus ou non explicites.

Nous avons su modéliser les grandes fonctionnalités du système. Cependant le système de relance en cas de conflit n'a pas été faite faute de temps. En effet, on aurait pu prévoir une fonctionnalité, qui fonctionnerait comme une alarme et qui permettrait de vérifier les échéances de paiement.

Certaines contraintes ne sont pas gérées par le système :

- L'ancienneté des acheteurs qui peut ouvrir droit à certains privilèges.
- La compétence des experts qui doit être prise en compte lors des choix des experts.
- La considération des mauvais payeurs et des bons payeurs. Le système aurait pu vérifier les échéances et modifier l'état des personnes c'est-à-dire qu'un règlement qui arrive avant la date échéance entraînerait la modification de l'état de la personne qui passera à « mauvais payeur ».
- La réputation des vendeurs.

En effet, le système se contente de stocker ces infos ; l'organisateur devra les gérer lui même et les consulter pour prendre ses décisions.

#### Il y a également une ambiguïté sur la modélisation de l'héritage....

En effet le fait d'avoir dans la tables Personne un attribut « type » fait qu'une personne ne peut pas avoir plusieurs types (exemple : personne acheteur et vendeur) contrairement à la réalité.

Nous avons donc tenu compte de la réalité dans notre spécification, cependant nous nous posons des questions sur l'utilité de l'attribut « type ». En effet notre héritage entre Personne Expert Acheteur et Vendeur l'exprime déjà.

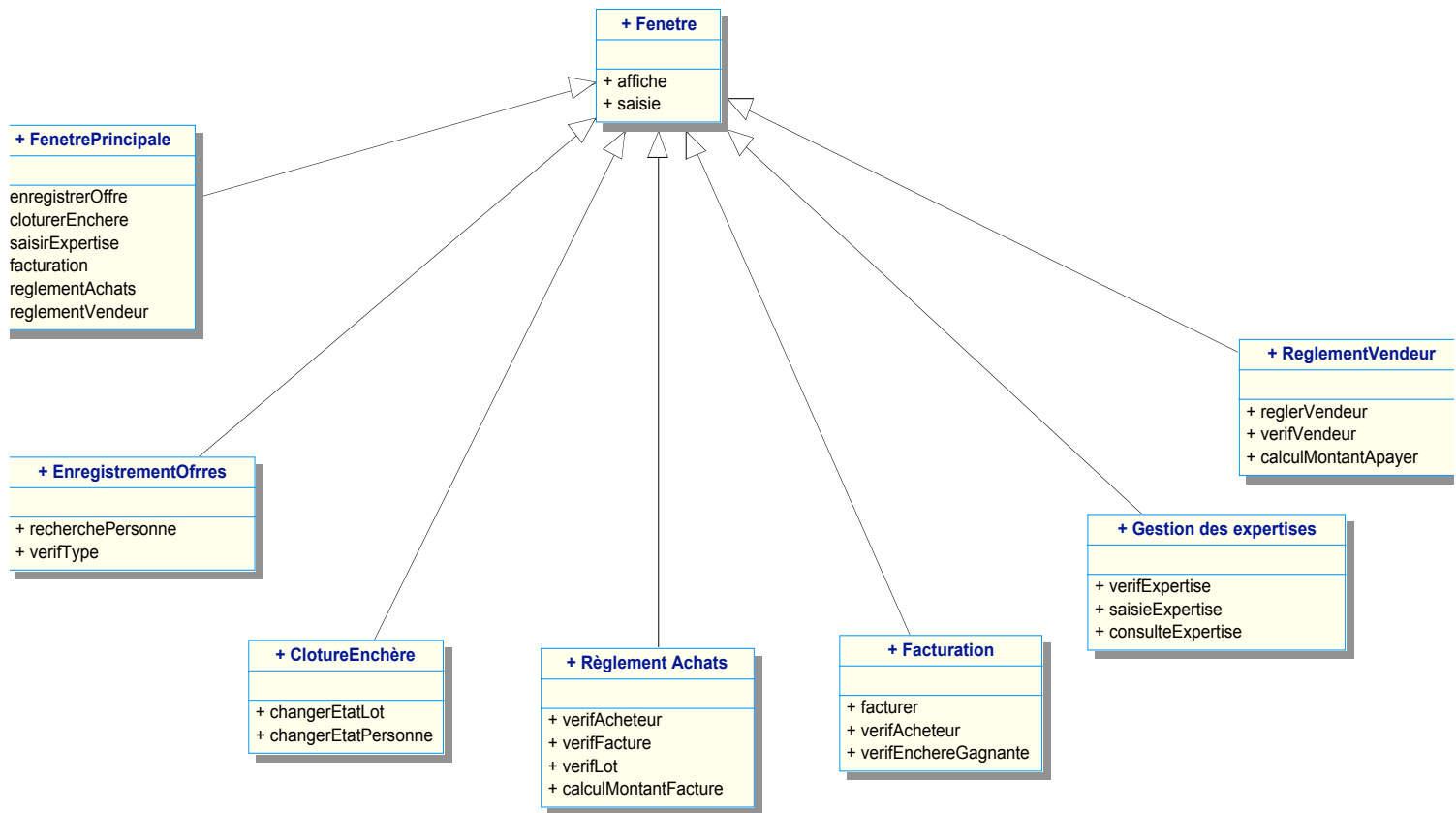
La suppression de l'attribut « type » entraînerait des modifications dans nos diagrammes de séquences.

Pour connaître le type d'une personne il faudrait interroger les ensembles correspondants au lieu de faire un test sur le type de la personne dans l'ensemble personne.

L'enregistrement d'une offre se ferait donc de la façon suivante :

Nous avons remarqué au final que les opérations de nos IHM n'étaient pas explicitement indiquées dans les diagrammes. En effet, les opérations d'affichage et de saisies n'avaient pas de méthodes correspondantes de même que les tests et traitements faits par l'interface.

Les changements au niveau du diagramme des classes pour les IHM seraient :

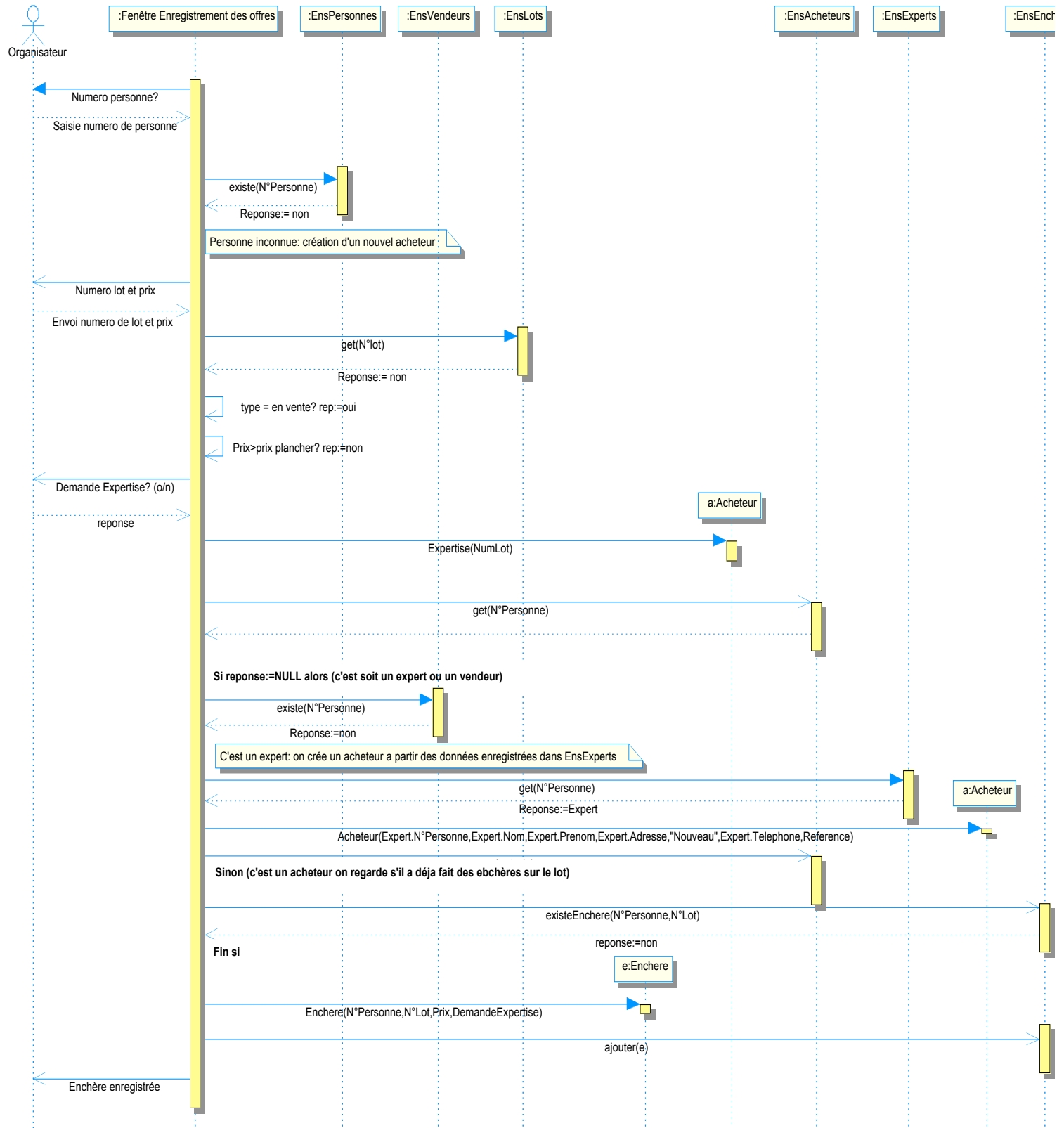


On a également ajouté une classe « FenetrePrincipale » qui présente le menu pour accéder aux différentes fonctionnalités du système.

Pour le cas enregistrements des offres des personnes non vendeurs il aurait fallu le diagramme de séquence à la page suivante.

Note : **EnsPersonnes** contient toutes les personnes confondues.

:enregistrement des offres: type personne <> vendeur		
:		
: : FAGBEGNON		
n : 1	Créé le : 21/10/2006	Modifié le : 8/11/2006



### **III. Conclusion**

Les différentes tâches fixées ont été réalisées à partir de plusieurs hypothèses. Nous avons modélisé les opérations importantes, respecté des contraintes fixées et modéliser la « vie » de nos classes à travers des automates.

Nous avons rencontré quelques difficultés qui nous ont amené à faire certains choix. Nous avons suivi la démarche de la modélisation UML, en respectant au mieux les formalismes cependant certains modèles peuvent être améliorés ou plutôt raffinés.

Par manque de temps nous n'avons pas pu développer certaines idées que nous avions. Grâce à ce projet nous avons pu mesurer les possibilités offertes par UML. C'est une méthode d'analyse riche mais certaines notations n'étant pas uniformisées ; la façon de modéliser une fonctionnalité peut être différente d'une personne à l'autre. Cependant il suit un principe assez intuitif qui permet d'exprimer clairement les étapes de réalisation d'un traitement.

Il aurait été intéressant d'implémenter le système à partir de notre spécification afin de mieux mesurer l'efficacité d'UML et des méthodes utilisées pour résoudre certains problèmes.