



Projet PSTL

Implantation d'un système de gestion d'enchères en BCM4Java



Encadrant : Jacques MALENFANT

Malik BOUAOUD - 21105730

Sandrine EAR - 21110030

Table des matières

Introduction	3
Conception du projet	5
Protocoles d'enchères	5
Diagrammes d'activités	5
Enchère à l'offre scellée	5
Enchère à l'anglaise	7
Diagramme sous la forme d'automates	9
Automate d'état fini de l'enchère à l'offre scellée	10
Automate d'état fini de l'enchère à l'anglaise	10
Diagramme de composants	11
Élaboration	12
Implémentation synchrone	13
Implémentation asynchrone	14
Greffons	16
Répartition	16
Modèles de décision : comportement des acheteurs	17
Scénarios de tests	18
Conclusion	22
Références	23

Introduction

Notre projet PSTL porte sur la réalisation d'un système de gestion d'enchères de vente, d'achat de biens ou de services en utilisant des concepts de la programmation par composants en Java avec la bibliothèque BCM4Java.

Ce concept de programmation met l'accent sur la construction d'une application réutilisable grâce aux composants, ce qui va permettre au système d'enchère d'être réutilisable en se greffant sur différents domaines d'application.

Un composant représente un objet Java décrit par une classe avec des méthodes définissant ses services. La notion de composants implique la présence des interfaces de composants, des ports, des connecteurs, de problèmes de concurrences et de parallélismes, ainsi que de leur répartition.

Les interfaces de composants sont des interfaces Java déclarant les signatures des méthodes appelables. Ces composants sont connectés entre eux à travers des connecteurs composés d'un port entrant et de son port sortant. Ce sont également des objets Java créés et possédés par les composants.

La création des composants est possible grâce au framework BCM4Java. Un composant peut requérir ou offrir des interfaces. Offrir une interface symbolise qu'un composant offre ses services à un autre. En revanche, requérir une interface signifie que pour exécuter certaines tâches, un composant utilise des services externes.

Ce système doit être automatisé dans une application dite autonome, ce qui signifie qu'il ne nécessite pas une intervention de la part des humains pour faire avancer l'enchère. En effet, cette réalisation ne se repose pas sur une intervention humaine comme pour le déroulement d'une enchère classique telle que nous la connaissons. Pour cela, il doit être également basé sur des modèles de décisions, mentionnée plus loin dans le rapport.

Notre implémentation repose sur la conception d'un système comportant trois rôles principaux. Les vendeurs peuvent offrir des biens ou des services en fixant un prix de référence permettant à des acheteurs potentiels d'enchérir avec une offre supérieure à celle-ci. Le commissaire-priseur est quant à lui chargé de la gestion de l'enchère, c'est-à-dire de récupérer les offres provenant des différents acheteurs et désigner le vainqueur de l'enchère lorsque celle-ci est terminée en indiquant le succès ou bien l'échec de la vente.

Il existe différents types de protocoles d'enchères dont l'enchère à l'anglaise et l'offre scellée.

L'enchère à l'anglaise possède un prix de référence minimum fixé par le vendeur de l'objet mis aux enchères. Les enchérisseurs peuvent émettre des offres supérieures au prix fixé et l'enchère s'arrête au moment où plus aucun participant ne propose un renchérissement.

L'offre scellée comporte uniquement un tour d'enchérissement où le commissaire-priseur reçoit les offres et lorsqu'il a fini de réceptionner toutes les offres des participants, il déterminera la meilleure offre.

Ce système d'enchères pourra être appliqué dans un contexte où les habitants au sein d'un quartier effectuent des échanges autour de l'électricité autoproduite.

Les habitants qui ont en leur possession trop d'énergie dans leur stockage peuvent mettre en enchère ce surplus. Leur but est de vendre leur énergie au prix le plus élevé possible. Les habitants qui souhaiteraient acheter de l'énergie ont quant à eux pour objectif d'acquérir de l'énergie supplémentaire au prix le plus bas.

Conception du projet

Protocoles d'enchères

Lors de la conception de notre projet, nous avons implémenté l'enchère à l'anglaise ainsi que l'enchère à l'offre scellée.

Avant de réaliser la partie code du sujet, nous avons commencé par l'élaboration de nos protocoles d'enchères à l'aide de différents types de diagrammes tels que des diagrammes d'activités, des diagrammes avec des automates ou bien encore des diagrammes à base de composants.

Un protocole est un mécanisme permettant la communication entre les différents agents présents dans notre système d'enchères ainsi que la définition des étapes des déroulements des enchères.

Comme nous l'avons mentionné dans la partie introductive, nous avons identifié trois rôles. Les rôles décrivent des comportements au cours des échanges et chacun des rôles peut réaliser des actions du protocole par exemple réaliser un sur enchérissement d'un bien pour l'acheteur.

Au départ, les rôles sont représentés sous forme de composants mais en avançant dans le projet, ces rôles vont être associés à des composants en installant des greffons selon les besoins.

Diagrammes d'activités

Les diagrammes d'activités permettent de visualiser les différentes étapes/actions réalisées au cours d'une enchère entre les différents acteurs qui dans notre cas sont le commissaire-priseur, les vendeurs et les acheteurs.

Sur les diagrammes d'activités que nous allons vous présenter, nous y distinguerons trois colonnes pour chacun des rôles du protocole. Dans ces colonnes, nous y retrouvons la description des actions de chacun d'entre eux pour le protocole décrit.

Enchère à l'offre scellée

Voici le diagramme d'activité correspondant à notre protocole d'enchère à l'offre scellée :

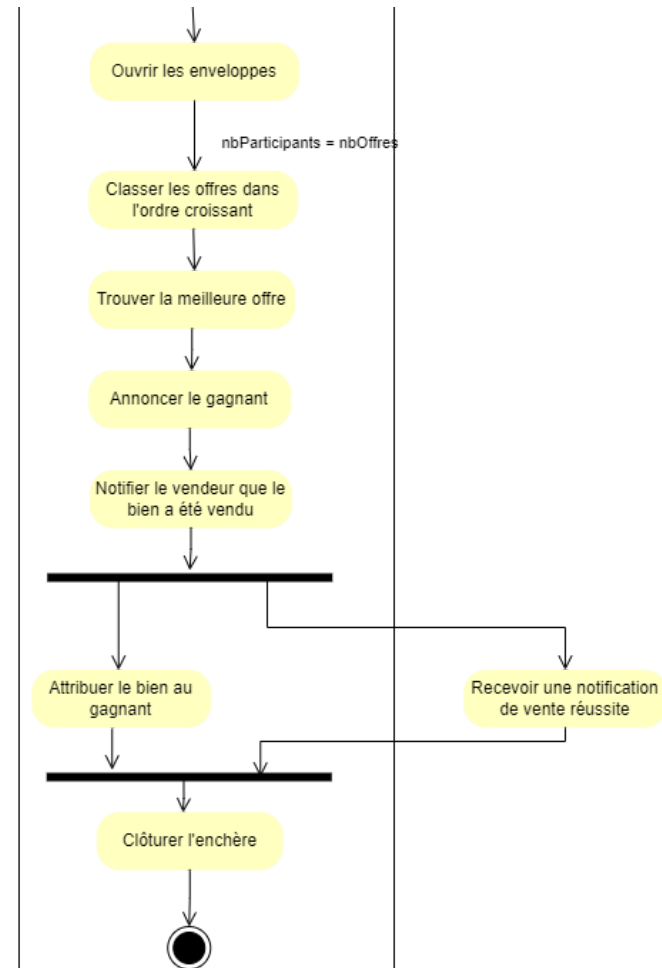
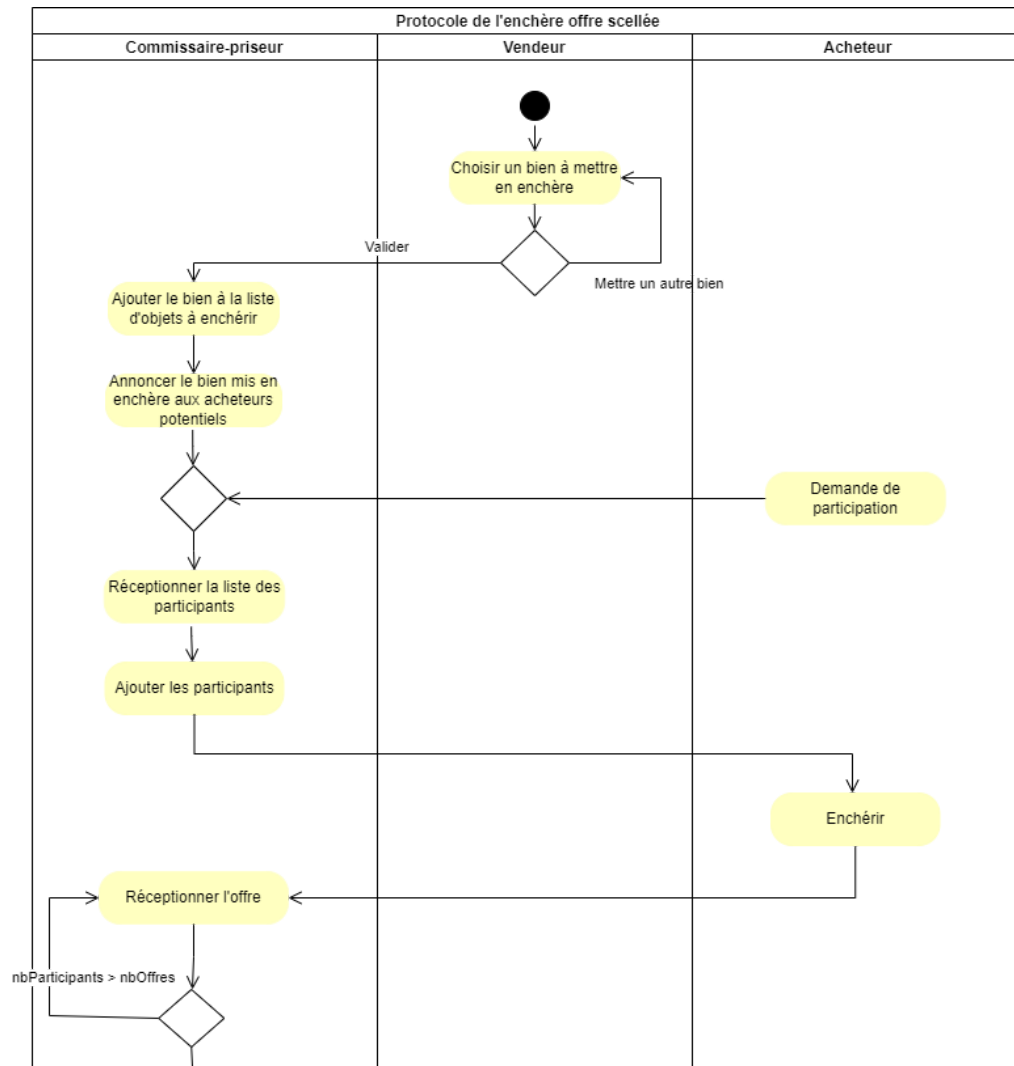


Figure 1 - Diagramme d'activité de l'enchère à l'offre scellée

Une enchère à l'offre scellée est composée d'un unique tour dans lequel les enchérisseurs peuvent émettre une somme pour enchérir l'objet. Le gagnant est celui qui a fait la meilleure offre.

Préparation de la mise en enchère

Afin de respecter et simuler au maximum le protocole d'offre scellée concrète au monde réel, nous avons décidé de ne pas mettre un prix de référence minimal mais il est tout de même possible de l'imposer.

Le rôle du vendeur consiste à choisir un objet à mettre en enchère en lui attribuant un nom, une description et envoie une demande de participation au commissaire-priseur.

Lorsque ce dernier la valide, le commissaire-priseur reçoit le bien à ajouter à la liste des objets à mettre en enchère et réalise une annonce de cet objet auprès des acheteurs potentiels.

À l'annonce de l'objet mis en enchère, les acheteurs potentiels vont répondre à l'invitation en émettant une requête de participation.

Les enchérisseurs ayant les moyens financiers souhaiteront participer à l'enchère, quant aux autres ils s'abstiendront.

Après avoir vérifié sa capacité d'accueil en filtrant les participants non admis à défaut de cette dernière, le commissaire-priseur aura en sa possession la liste de tous les participants et l'enchère avec le protocole à l'offre scellée peut donc commencer.

Déroulement d'un tour (enchère)

Comme mentionné ci-dessus, l'offre scellée se déroule en un seul tour, une fois la préparation de l'enchère faite. Les acheteurs font une unique offre sous forme d'enveloppe scellée, c'est-à-dire que les autres participants de l'enchère n'ont pas connaissance de la valeur émise et ne peuvent donc pas "surenchérir".

Clôture de l'enchère

Une fois que toutes les offres sont reçues, le commissaire-priseur dépouille les enveloppes, désigne le vainqueur et informe le vendeur que son bien a été vendu.

Enfin, l'enchère est clôturée.

Enchère à l'anglaise

Voici le diagramme d'activité correspondant à notre protocole d'enchère à l'anglaise :

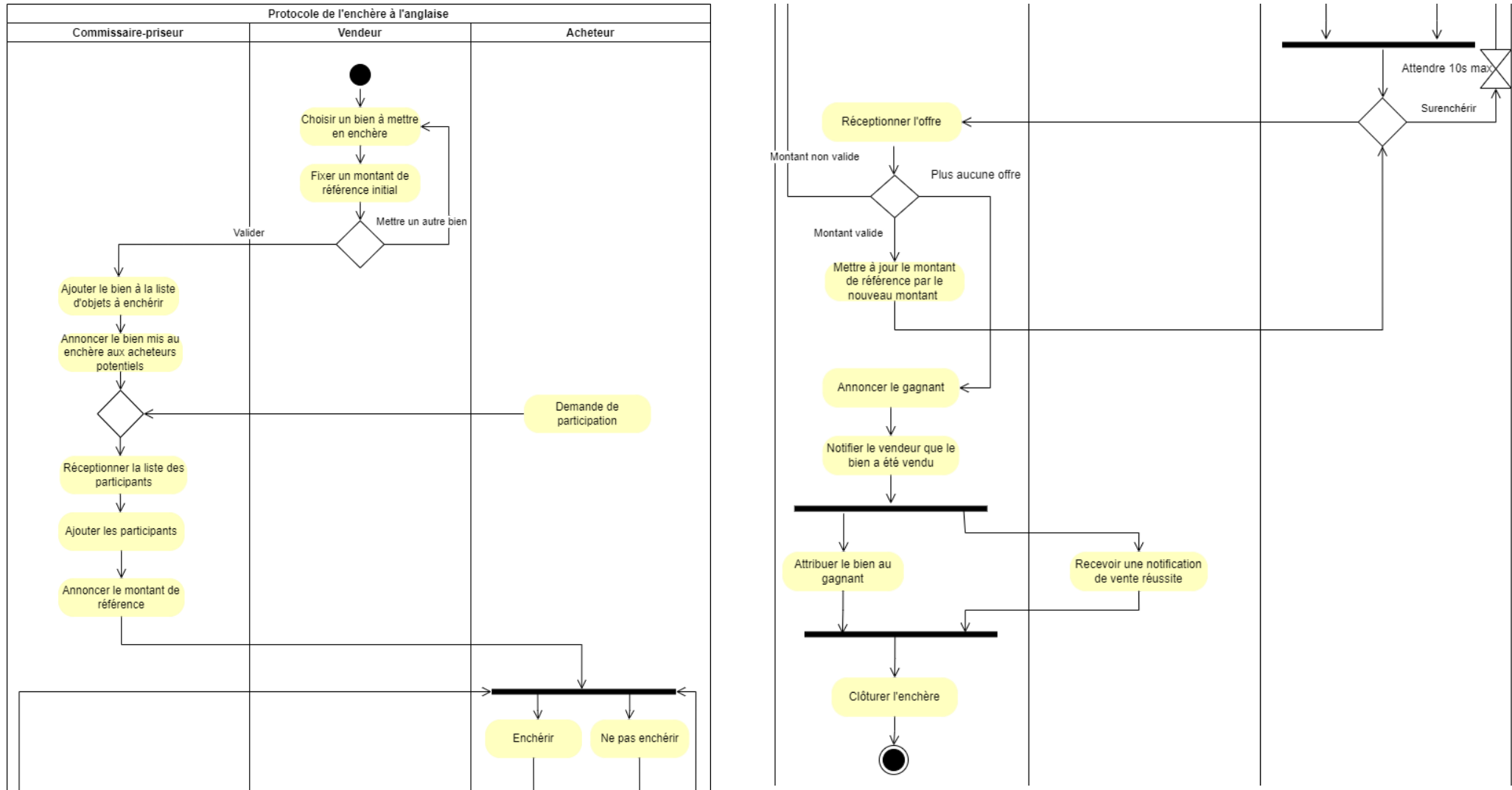


Figure 2 - Diagramme d'activité de l'enchère à l'anglaise

Pour faire un rappel, l'enchère à l'anglaise n'est pas clôturée tant qu'il existe un participant qui émet des offres. Ce protocole peut être réalisé de deux façons différentes notamment une enchère ascendante ou bien descendante.

Préparation de la mise en enchère

La partie préparation de la mise en enchère est semblable à celle de l'offre scellée, cependant, dans l'enchère à l'anglaise, le protocole s'appuie sur l'utilisation d'un prix de référence initial sur lequel les participants s'appuient pour surenchérir le bien.

Déroulement de l'enchère à l'anglaise

Pour notre protocole, nous avons décidé de partir sur une enchère ascendante avec un prix de référence minimal fixé par le vendeur de l'objet.

Le commissaire-priseur commence l'enchère par annoncer à tous les participants le prix de référence et le met à jour au fur et à mesure que l'enchère augmente.

Tant que les acheteurs sont intéressés par le bien et peuvent poursuivre l'enchère, ils enchérissent sinon ils se rétractent.

Clôture de l'enchère

Lorsque plus aucune offre n'est émise au bout d'un délai prédéfini, le commissaire-priseur peut alors clôturer l'enchère en annonçant le gagnant, c'est-à-dire la dernière offre valide reçue. L'enchère est ensuite clôturée.

Diagramme sous la forme d'automates

Quant à l'utilisation des diagrammes sous la forme d'automates, ils permettent de visualiser les différents états dans lequel nous pouvons nous retrouver au cours du déroulement d'une enchère. Ces diagrammes permettent de faire apparaître les enchaînements des états par lesquels passe chaque rôle et ainsi conserver les données utiles tout au long du déroulement du protocole.

Les états de l'automate représentent l'avancée dans l'enchère, quant aux transitions entre les états, elles font référence aux différentes opérations réalisées durant notre protocole d'enchère.

Automate d'état fini de l'enchère à l'offre scellée

Le diagramme ci-dessous représente les états rencontrés au cours d'une enchère à l'offre scellée.

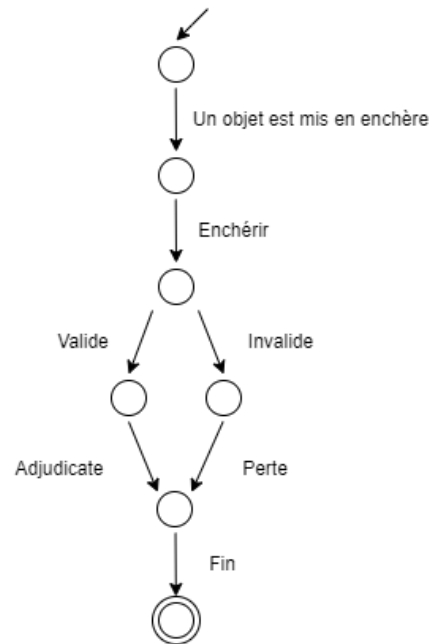


Figure 3 - Automate d'état fini de l'enchère à l'offre scellée

L'automate de l'offre scellée est assez simple à comprendre puisque l'acheteur a le droit de faire une unique offre.

Automate d'état fini de l'enchère à l'anglaise

Le diagramme suivant illustre les états pour une enchère suivant le protocole de l'enchère à l'anglaise.

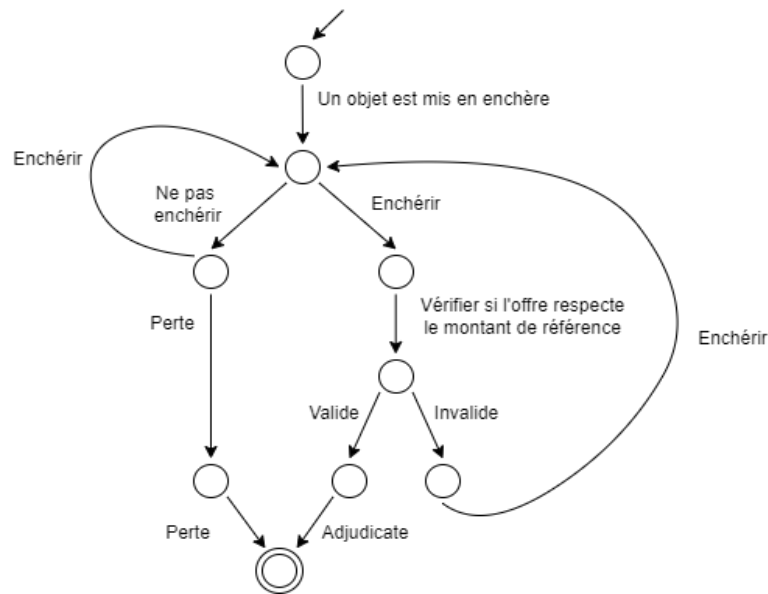


Figure 4 - Automate d'état fini de l'enchère à l'anglaise

Sur ce diagramme, nous pouvons remarquer la présence d'un circuit contrairement à l'automate précédent, qui représente la boucle sur laquelle les enchérisseurs font leurs offres jusqu'à l'abstention.

Ces automates nous ont permis de mieux aborder la partie conception des protocoles de l'enchère, notamment en identifiant les méthodes à créer et les changements d'état à réaliser durant un protocole d'enchère.

Diagramme de composants

L'architecture de notre projet repose sur une approche à base de composants. Comme il a été précisé dans le cahier des charges.

Pour réaliser un protocole d'enchères, il nous a fallu identifier les différents rôles pour une enchère ainsi que leurs actions. Nous avons notamment distingué trois rôles principaux représentés sous forme de composants.

Dans les architectures à base de composants, les interfaces offertes et requises représentent les échanges entre ces derniers permettant ainsi de réaliser le diagramme de composants suivant :

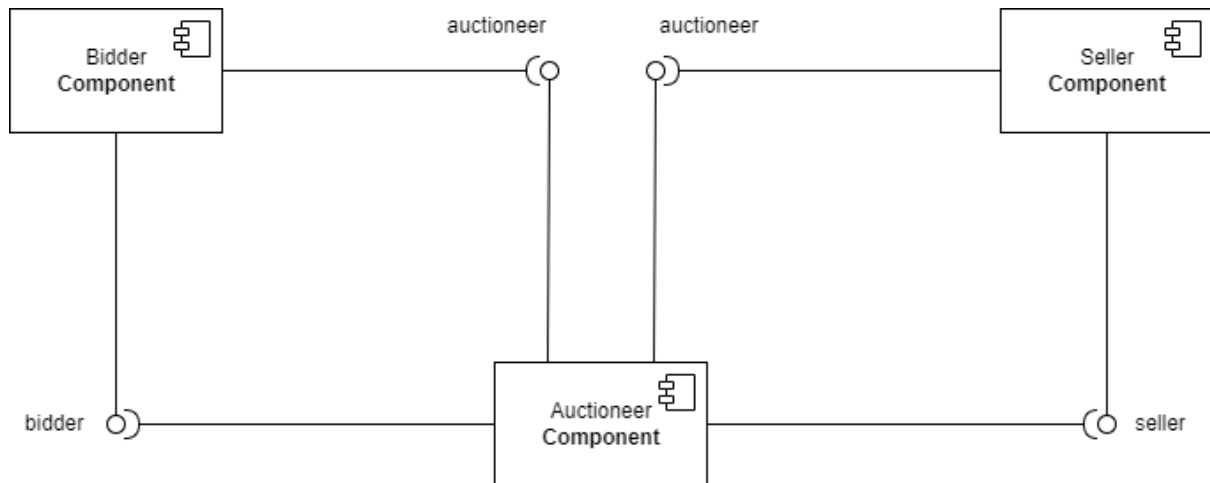


Figure 5 - Diagramme de composants

Chaque composant offre sa propre interface, c'est-à-dire qu'un composant Seller offre une interface Seller permettant de former des briques réutilisables par les autres composants. Pour les interfaces requises, elles représentent les méthodes requises à un composant afin d'offrir ses différents services.

Le composant Seller requiert Auctioneer pour pouvoir envoyer une demande de vente.

Le composant Bidder a besoin du composant auctioneer pour émettre des offres et envoyer des requêtes de participation à l'enchère.

Le composant Auctioneer par exemple requiert tous les autres composants car pour réaliser une enchère, nous avons besoin à la fois d'un vendeur et de plusieurs acheteurs.

Toutefois, ce diagramme ne représente pas entièrement la réalité, c'est-à-dire qu'il peut exister plus de composants que ce qui est représenté. Par exemple, dans un vrai scénario d'enchères, nous pourrions avoir une dizaine d'acheteurs représentés par le composant Bidder.

Élaboration

L'élaboration du projet s'est déroulée en quatre étapes essentielles. Nous avons commencé par une première implémentation synchrone du système d'enchère, une seconde asynchrone, suivie de l'utilisation de greffons (plugins) et enfin le passage à une architecture répartie.

Implémentation synchrone

Réalisations

Une implémentation synchrone d'un protocole consiste à appeler un enchaînement d'actions dans l'ordre depuis le début de l'enchère jusqu'à sa clôture. Dans le contexte actuel de l'application, une solution synchrone symbolise le fait de dérouler qu'une seule et unique enchère à la fois.

Les diagrammes présentés ci-dessus nous ont facilité l'implémentation du code et nous donnent ainsi une meilleure visualisation de l'ensemble des étapes d'une enchère.

Nous avons réalisé notre projet en commençant par implémenter les interfaces et les classes Java en se référant aux différents diagrammes d'activités présentés précédemment.

Tout d'abord, nous avons codé de manière synchrone pour implémenter les protocoles. Afin de tester le code écrit, nous avons réalisé des tests unitaires pour vérifier le bon fonctionnement des différentes méthodes.

Ensuite, nous avons intégré les composants à notre projet en considérant les rôles c'est-à-dire le commissaire-priseur, le vendeur, l'acheteur comme des composants.

L'acheteur est représenté par le composant Bidder, le commissaire-priseur par le composant Auctioneer et enfin le composant Seller pour la représentation d'un vendeur. Les différents composants se distinguent parmi les autres grâce aux méthodes qu'ils implémentent et les interfaces qu'ils offrent ou qu'ils requièrent. Prenons par exemple le composant Bidder, il requiert une interface Auctioneer, puisque ce dernier doit envoyer une requête de participation au commissaire-priseur pour participer à l'enchère.

La création des ports et des connecteurs a été basée sur le diagramme des composants présentés plus haut.

Difficultés rencontrées

Dans un premier temps, nous avons développé notre application sur une base Java normale, c'est-à-dire sans une architecture à base de composants dans le but de mieux appréhender le problème principal qui était de créer un système d'enchère complet.

N'ayant jamais travaillé sur une architecture à base de composants auparavant, l'utilisation et la compréhension du framework BCM4Java était un élément délicat et fastidieux.

Solutions trouvées

Pour surmonter cette difficulté, nous nous sommes appuyés sur des projets basés sur BCM4Java fourni par notre encadrant qui nous a permis de transformer nos classes Java en composant avec toutes leurs interfaces offertes, requises et leurs connecteurs.

Implémentation asynchrone

Le passage du projet d'une implémentation synchrone à une implémentation asynchrone implique une introduction du parallélisme et de la gestion de la concurrence.

Une implémentation asynchrone ne considère plus des enchaînements d'actions séquentielles de la même façon qu'en synchrone, puisque cette solution permet de gérer plusieurs enchères à la fois.

Réalisations

L'introduction du parallélisme dans notre système d'enchères consistait à utiliser des threads dans une architecture centralisée afin que le composant commissaire-priseur puisse être en mesure de gérer correctement plusieurs enchères en même temps.

Pour réaliser cela, nous avons d'abord modifié nos ports entrants du composant Auctioneer et changé les méthodes d'appel synchrone en appels asynchrones pour supporter l'utilisation des threads. Les appels synchrones bloquent le thread appelant tant que celui-ci ne reçoit pas un résultat de retour et ne peut donc pas poursuivre son exécution. Les appels asynchrones quant à eux libèrent immédiatement le thread de l'appelant et ce thread poursuit son exécution. Cependant, ce changement a été opéré uniquement sur les méthodes participant au déroulement d'une enchère.

Ensuite, nous avons créé des pools de threads. Ils représentent un groupe de processus légers appartenant au commissaire-priseur permettant de lancer différentes enchères à la fois, tout en vérifiant si les différentes étapes de l'enchère se sont bien déroulées.

Difficultés rencontrées

Lors de cette étape, nous avons été confrontés à plusieurs écueils "célestres" du monde de la concurrence.

Nous pouvons notamment citer les sections critiques qui définissent une zone où la ressource ne peut être utilisée que par un seul processus. Il existe également le problème de compétition (Race) lors de l'accès simultané à la même ressource ou bien encore l'interblocage (Deadlock) lorsqu'il existe une séquence d'entrelacement où des processus restent bloqués et leurs tâches sont donc non terminées.

Une première approche consistait à gérer plusieurs enchères à l'aide de structures de données partagées, c'est-à-dire qu'un commissaire-priseur centralise toutes les informations dans les mêmes structures de données et ceux quelle que soit l'enchère.

En complément du problème de compétition lors de l'accès à cette structure, nous perdions des informations relatives à une enchère donnée. Dans ce contexte précis, nous avons une structure de données de classement qui contenait les meilleures offres des acheteurs ainsi que

leurs montants centralisés dans un seul HashMap. Toutefois, avec la présence de plusieurs enchères, nous n'avions plus les moyens de savoir si une offre a été émise pour un objet précis d'une enchère donnée.

L'introduction des processus légers dans notre projet a de plus causé la perte de l'ordonnancement de certaines tâches notamment si nous plaçons chaque étape de l'enchère dans un processus, nous devons faire en sorte que l'enchaînement soit correct.

Par exemple, si un thread venait à clôturer l'enchère après un certain temps, l'ensemble des offres émises devrait s'arrêter. Donc, cela impliquerait qu'une offre émise en cours de la clôture d'une enchère doit être rejetée.

Solutions trouvées

Pour pouvoir suivre l'état d'avancement de plusieurs enchères à la fois, nous avons créé un "état d'enchère" permettant de suivre la progression d'une seule enchère quel que soit son protocole (offre scellée ou anglaise), de connaître l'objet mis en adjudication et la création d'un classement pour les enchérisseurs.

Grâce à cette structure, nous avons pu éviter la section critique présente lors de la perte d'information liée à l'enchère lorsqu'un commissaire-priseur en déroule plusieurs à la fois.

Pour régler un autre problème de concurrence précédemment cité, nous avons utilisé à l'intérieur des états d'enchères des structures de données concurrentes qui résistent à plusieurs appels de threads tels que les ConcurrentHashMap à la place des HashMap habituelles ou bien des Vector au lieu d'ArrayList. Ainsi, nous n'avons plus d'écueil de compétition.

Afin de vérifier le passage par tous les états (cf diagramme automates) de l'enchère et que chaque méthode est appelée au bon moment, nous avons fait en sorte que les méthodes provoquent un changement d'état courant dans le protocole d'enchère exécuté.

Par exemple, à la fin de l'étape de "préparation de l'enchère", l'état de l'enchère passe de "initialisation" à "enchère en cours".

Nos protocoles d'adjudications se déroulent durant un certain temps. En BCM4Java, nous l'appelons la durée de vie des composants représentés sous la forme d'un cycle. Une fois cette durée de vie atteinte ou que la méthode de clôture d'enchères a été enclenché, nous devons arrêter toutes les offres émises. Pour cela, nous avons utilisé un mutex, un système d'exclusion mutuelle binaire en d'autres termes un sémaphore binaire, pour garantir que si la méthode de fermeture est lancée, toutes les autres ne peuvent plus s'exécuter et ainsi mettre fin aux enchères.

Greffons

Une implémentation sans greffons implique que les différents rôles d'une enchère sont adoptés par les composants selon leurs besoins. L'ajout des greffons dans un système d'enchères va alors permettre aux rôles d'être liés directement à des greffons.

En BCM, les greffons font appel à des objets délégués dans les composants pour une réutilisation du code qui se trouvent dans le plugin. Donc, une telle implémentation permet de factoriser le code et d'avoir une meilleure réutilisation du code.

En Java, l'intégration d'un greffon au sein d'un projet nécessite le respect du cycle de vie d'un greffon composé de quatre méthodes en respectant un ordre précis.

Dans notre projet, les plugins remplacent les ports entrants, c'est-à-dire les interfaces requises des composants. Par exemple, pour le composant Auctioneer, le port offrant sa propre interface a été transformé en plugin. De même pour les composants Bidder et Seller, Nous pourrions aussi avoir un commissaire-priseur adoptant le rôle d'un vendeur uniquement en installant le greffon du vendeur.

Répartition

La programmation répartie est une exécution sur plusieurs unités de traitement indépendantes (Les ordinateurs, Java Virtual Machines, etc ...). En BCM4Java, la programmation par composant répartie autorise les appels entre composants distants et s'appuie sur une technologie Java/RMI, un appel de procédures à distance en appelant une procédure entre deux processus. Nous pouvons faire exécuter du code en provenance de différentes JVM.

L'implantation de la répartition des composants Auctioneer, Bidder et Seller permet de les rendre déployables sur plusieurs JVM et sont également interconnectables entre eux.

En général, dans les ventes aux enchères, le commissaire-priseur représente le mandataire entre les vendeurs et les acheteurs. C'est pourquoi nous avons décidé de répartir notre système d'enchère entre deux JVM avec une répartition des composants de la manière suivante. Cette répartition n'est pas uniquement la seule façon de faire, par exemple, nous aurions pu déplacer le composant Bidder dans une troisième JVM.

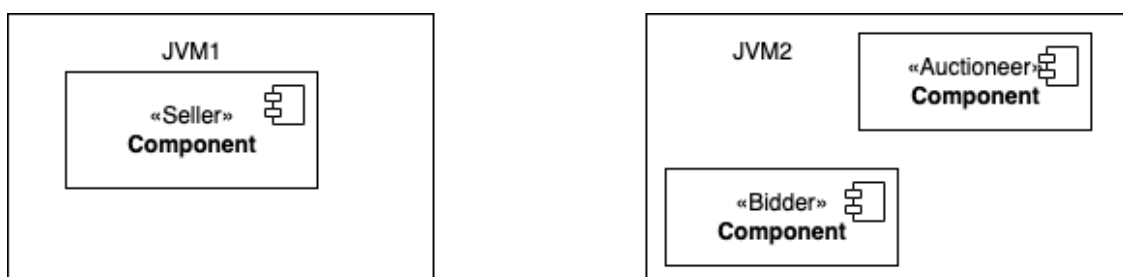


Figure 6 - Schéma de l'architecture répartie

Un composant Seller envoie des demandes de ventes à partir de la JVM1 vers l'auctioneer de la JVM2.

Au premier abord, on peut apparenter un tel schéma à une programmation parallèle multi-threadée. Cependant, des différences significatives sont à noter notamment lors du passage d'informations d'une JVM à une autre.

Même si BCM masque les mécanismes JAVA afin de simplifier la programmation répartie, nous avons appris les principes d'appel de méthode à distance dans la technologie RMI ainsi que leurs déroulements.

L'appel à des méthodes à distance RMI étant les procédures RPC qui consiste à utiliser deux processus, un processus appelant et un processus appelé.

Ces procédures reposent sur un échange via sockets qui suppose un mécanisme de sérialisation et désérialisation, en d'autres termes un encodage de valeurs. Nous avons donc fait en sorte que les objets circulant entre les JVMs soient tous sérialisables.

Le passage à une architecture répartie repose aussi sur la synchronisation des cycles de vie des composants se trouvant sur des JVMs différentes. Par exemple, un composant n'ayant pas encore commencé sa construction ne peut pas recevoir d'appel venant d'autres composants. Pour cela, nous avons utilisé la barrière cyclique offerte par BCM pour permettre une telle synchronisation.

Un autre aspect d'une telle approche est le stockage des différents ports offerts et requis dans des registres. Au sein d'une JVM, nous avons ce qu'on appelle des registres locaux contenant tous les URIs (identifiant unique d'un port) des composants présents dans la JVM. Mais lors du passage à un système réparti, nous devons passer par un registre global qui retourne la JVM hôte du composant recherché, ce qui nous permet d'accéder au bon registre local et tout cela grâce au mécanisme Global Registry offert par BCM.

Modèles de décision : comportement des acheteurs

Un des aspects de ce projet consistait à élaborer un certain type de comportement pour les acheteurs. En clair, nous devons répondre à différentes problématiques comme par exemple, quand est-ce qu'un enchérisseur décide de surenchérir, de se retirer ou alors quand est-ce qu'une personne est vraiment intéressée par le bien mis en enchère ... etc.

Pour cela, nous nous sommes inspirés d'un article universitaire nommé Bidding behavior in competing auctions: Evidence from eBay [1] trouvé sur la base de données PRIMO de Sorbonne Université. Cet article présente les différents types d'acheteur et leurs façons d'enchérir. Il explicite également le fait que si un acheteur a décidé d'enchérir, alors il aura plus de chances de surenchérir à l'instant suivant.

Dans notre application, nous avons essayé de simuler un comportement similaire en distinguant trois catégories d'acheteur: agressive, active et passive.

Les différentes catégories reflètent le comportement que va avoir un acheteur potentiel. Un acheteur adoptant un comportement agressif surenchérit avec une hausse de 60% par rapport au montant actuel de l'enchère, puis de 40% et de 10% pour les acheteurs actifs et passifs respectivement.

Pour mieux représenter le contexte réel d'une enchère, nous avons aussi pris en compte les moyens financiers d'un enchérisseur en fixant des limites à ne pas dépasser lors d'un sur-enchérissement.

Ce critère autour des moyens d'un acheteur potentiel nous a permis de bien simuler le déroulement d'une enchère tout en respectant les contraintes de la vie réelle mais également avoir des jeux de tests plus intéressants à la fin. Bien évidemment, le comportement des acheteurs a seulement été intégré dans le protocole d'enchère à l'anglaise puisque dans les offres scellées l'intéressé n'émet qu'une seule et unique offre.

Scénarios de tests

La première simulation des tests d'intégration sur les deux types d'enchères intégrés à notre projet s'est réalisée de manière séparée, c'est-à-dire que nous avons créé deux CVM, une pour l'enchère à l'anglaise et une autre pour l'enchère à l'offre scellée. Cette simulation a été réalisée sur une mono-JVM avant le passage au déploiement sur plusieurs JVM.

Pour tester notre système d'enchères, nous avons créé deux types de tests.

Un premier sur une enchère à l'offre scellée, le composant Seller met en enchère trois unités énergétiques différentes grâce à une demande de vente valant 100, 200 et 300 euros respectivement.

Le composant commissaire-priseur met les objets reçus en enchère en les plaçant dans différents threads. Les composants acheteurs prennent en paramètre l'objet convoité et soumettent des offres uniquement connues par le commissaire-priseur.

Afin de simuler au mieux le déroulement de l'enchère, nous avons utilisé la bibliothèque faker pour générer des prénoms à nos 30 acheteurs.

À l'adjudication, le vainqueur est annoncé avec un classement des différentes offres.

Voici un exemple d'affichage pour le déroulement d'une enchère à l'offre scellée :

```
starting...
executing...
A selling request of 6000 Kilos Watts of solar energy has been sent.
A selling request of 8000 Kilos Watts of solar energy has been sent.
A selling request of 3000 Kilos Watts of solar energy has been sent.
The selling request has been received and accepted.
The auction id {1}, 3000 Kilos Watts of solar energy start
---
The selling request has been received and accepted.
The selling request has been received and accepted.
```

```

The auction id {3}, 8000 Kilos Watts of solar energy start
---
The auction id {2}, 6000 Kilos Watts of solar energy start
---
Closed envelope of Ernest for {1} with the amount : 50.0€
Closed envelope of Marcos for {2} with the amount : 200.0€
Closed envelope of Duncan for {3} with the amount : 300.0€
Closed envelope of Aurelia for {1} with the amount : 100.0€
Closed envelope of Rona for {2} with the amount : 250.0€
Closed envelope of Katina for {3} with the amount : 350.0€
Closed envelope of Eldridge for {3} with the amount : 400.0€
Closed envelope of Paulita for {1} with the amount : 150.0€
Closed envelope of Howard for {2} with the amount : 300.0€
Closed envelope of Dana for {3} with the amount : 450.0€
Closed envelope of Clair for {1} with the amount : 200.0€
Closed envelope of Lyndon for {2} with the amount : 350.0€
Closed envelope of Andrew for {1} with the amount : 250.0€
Closed envelope of Tomiko for {2} with the amount : 400.0€
Closed envelope of Jewell for {3} with the amount : 500.0€
Closed envelope of Moshe for {1} with the amount : 300.0€
Closed envelope of Sadie for {3} with the amount : 550.0€
Closed envelope of Reuben for {2} with the amount : 450.0€
Closed envelope of Aaron for {1} with the amount : 350.0€
Closed envelope of Cherri for {2} with the amount : 500.0€
Closed envelope of Mohammed for {3} with the amount : 600.0€
Closed envelope of Georgianne for {3} with the amount : 650.0€
Closed envelope of Houston for {1} with the amount : 400.0€
Closed envelope of Wesley for {2} with the amount : 550.0€
Closed envelope of Ashton for {2} with the amount : 600.0€
Closed envelope of Rolando for {3} with the amount : 700.0€
Closed envelope of Hosea for {1} with the amount : 450.0€
Closed envelope of Sammy for {1} with the amount : 500.0€
Closed envelope of Federico for {2} with the amount : 650.0€
The winner of 8000 Kilos Watts of solar energy in the auction {3} is : 'Rolando' and made an offer of 700.0€
The seller of the threshold bidding has been notified.
Ranking {3}: {Duncan=300.0, Katina=350.0, Eldridge=400.0, Dana=450.0, Jewell=500.0, Sadie=550.0, Mohammed=600.0,
Georgianne=650.0, Rolando=700.0}
The threshold auction {3} is closed.
Thanks for participating.
Closed envelope of Augustine for {1} with the amount : 550.0€
The winner of 6000 Kilos Watts of solar energy in the auction {2} is : 'Federico' and made an offer of 650.0€
The seller of the threshold bidding has been notified.
Ranking {2}: {Marcos=200.0, Rona=250.0, Howard=300.0, Lyndon=350.0, Tomiko=400.0, Reuben=450.0, Cherri=500.0, Wesley=550.0,
Ashton=600.0, Federico=650.0}
The threshold auction {2} is closed.
Thanks for participating.
The winner of 3000 Kilos Watts of solar energy in the auction {1} is : 'Augustine' and made an offer of 550.0€
The seller of the threshold bidding has been notified.
Ranking {1}: {Ernest=50.0, Aurelia=100.0, Paulita=150.0, Clair=200.0, Andrew=250.0, Moshe=300.0, Aaron=350.0, Houston=400.0,
Hosea=450.0, Sammy=500.0, Augustine=550.0}
The threshold auction {1} is closed.
Thanks for participating.
finalising...
shutting down...
ending...

```

Même principe pour l'enchère à l'anglaise, sauf comme mentionné précédemment l'enchère ne se déroule pas en un tour et les acheteurs ont la possibilité de surenchérir.

Pour ce test, nous avons mis en enchère cinq objets différents avec une trentaine d'acheteur avec divers comportements (passif, actif ou agressif).

Les enchères commencent à 100 euros pour le premier objet et 300 euros pour le dernier.

Les acheteurs qui ne peuvent plus suivre l'enchère se retirent instantanément et les enchères sont closes dès qu'on a une adjudication.

Les scénarios de tests affichés ci-dessous détaillent ces explications.

```

starting...
executing...
The auction id {1}, 3000 Kilos Watts of solar energy start at 100.0€
---
The auction id {4}, 10000 Kilos Watts of solar energy start at 300.0€

```

 The auction id {2}, 6000 Kilos Watts of solar energy start at 200.0€

 The auction id {5}, 12000 Kilos Watts of solar energy start at 300.0€

 The auction id {3}, 9000 Kilos Watts of solar energy start at 300.0€

 Sharonda is retrieving from the auction.
 Sharonda is retrieving from the auction.
 Sharonda is retrieving from the auction.
 Carlene bid for {3} with 480.0€
 Is there another offer above 480.0€ for {3} ?
 Jeanmarie bid for {1} with 110.0€
 Jeanmarie is retrieving from the auction.
 Is there another offer above 110.0€ for {1} ?
 Tyler bid for {5} with 330.0€
 Stevie bid for {4} with 330.0€
 Is there another offer above 330.0€ for {5} ?
 Is there another offer above 330.0€ for {4} ?
 Golden bid for {2} with 280.0€
 Is there another offer above 280.0€ for {2} ?
 Jere is retrieving from the auction.
 Rudolph bid for {4} with 363.0€
 Is there another offer above 363.0€ for {4} ?
 Wendolyn bid for {3} with 768.0€
 Yvette bid for {2} with 392.0€
 Is there another offer above 768.0€ for {3} ?
 Is there another offer above 392.0€ for {2} ?
 Clark bid for {5} with 363.0€
 Astrid bid for {1} with 121.0€
 Is there another offer above 363.0€ for {5} ?
 Is there another offer above 121.0€ for {1} ?
 Yvette is retrieving from the auction.
 Maura bid for {3} with 1228.8€
 Is there another offer above 1228.8€ for {3} ?
 Kate bid for {2} with 548.8€
 Laurinda bid for {4} with 399.3€
 Is there another offer above 548.8€ for {2} ?
 Is there another offer above 399.3€ for {4} ?
 Tanner bid for {5} with 399.3€
 Is there another offer above 399.3€ for {5} ?
 Ina bid for {1} with 133.1€
 Is there another offer above 133.1€ for {1} ?
 Milo is retrieving from the auction.
 Ina is retrieving from the auction.
 Stan bid for {4} with 439.22998€
 Is there another offer above 439.22998€ for {4} ?
 Timika is retrieving from the auction.
 Randell bid for {2} with 768.32€
 Is there another offer above 768.32€ for {2} ?
 Kate is retrieving from the auction.
 Tyree bid for {3} with 1966.0801€
 Is there another offer above 1966.0801€ for {3} ?
 Bertram bid for {1} with 146.41€
 Lashanda bid for {5} with 439.22998€
 Is there another offer above 146.41€ for {1} ?
 Is there another offer above 439.22998€ for {5} ?
 Stan is retrieving from the auction.
 Debbi bid for {4} with 483.15298€
 Caleb is retrieving from the auction.
 Is there another offer above 483.15298€ for {4} ?
 Shalonda is retrieving from the auction.
 Myron bid for {2} with 1075.648€
 Is there another offer above 1075.648€ for {2} ?
 Bertram is retrieving from the auction.
 Randell is retrieving from the auction.
 Raquel is retrieving from the auction.
 Elenore is retrieving from the auction.
 Myron is retrieving from the auction.
 Helga bid for {3} with 3145.728€
 Is there another offer above 3145.728€ for {3} ?
 Garrett bid for {1} with 161.05101€
 Wilton bid for {5} with 483.15298€
 Is there another offer above 161.05101€ for {1} ?
 Is there another offer above 483.15298€ for {5} ?
 Toshiko is retrieving from the auction.
 Enoch is retrieving from the auction.
 Asa bid for {2} with 1505.9071€
 Is there another offer above 1505.9071€ for {2} ?
 Devin bid for {4} with 531.46826€
 Is there another offer above 531.46826€ for {4} ?
 Dorethea is retrieving from the auction.
 Devon is retrieving from the auction.
 Asa is retrieving from the auction.
 Tod is retrieving from the auction.

Rana is retrieving from the auction.
 Golden is retrieving from the auction.
 Rudolph is retrieving from the auction.
 Kelsie is retrieving from the auction.
 Maura is retrieving from the auction.
 Astrid is retrieving from the auction.
 Laurinda is retrieving from the auction.
 Tyree is retrieving from the auction.
 Lashanda is retrieving from the auction.
 Monroe is retrieving from the auction.
 Wilton is retrieving from the auction.
 Helga is retrieving from the auction.
 Perry is retrieving from the auction.
 Devin is retrieving from the auction.
 Marcella is retrieving from the auction.
 Carlene is retrieving from the auction.
 Wendolyn is retrieving from the auction.
 Tyler is retrieving from the auction.
 Tanner is retrieving from the auction.
 Debbi is retrieving from the auction.
 Garrett is retrieving from the auction.
 Akiko is retrieving from the auction.
 Stevie is retrieving from the auction.
 Clark is retrieving from the auction.
 Pamila is retrieving from the auction.
 Fabian is retrieving from the auction.
 Toby is retrieving from the auction.
 Aron is retrieving from the auction.
 Francine is retrieving from the auction.
 Oscar bid for {2} with 2108.27€
 Is there another offer above 2108.27€ for {2} ?
 Latina bid for {2} with 2951.5781€
 Is there another offer above 2951.5781€ for {2} ?
 Oscar bid for {2} with 4132.2095€
 Is there another offer above 4132.2095€ for {2} ?
 Latina bid for {2} with 5785.0933€
 Is there another offer above 5785.0933€ for {2} ?
 Oscar is retrieving from the auction.
 Ranking {3}: {Carlene=480.0, Wendolyn=768.0, Maura=1228.8, Tyree=1966.0801, Helga=3145.728}
 The winner of the auction {3} is Helga and made an offer of 3145.728€
 Ranking {2}: {Golden=280.0, Yvette=392.0, Kate=548.8, Randell=768.32, Myron=1075.648, Asa=1505.9071, Oscar=4132.2095, Latina=5785.0933}
 Ranking {5}: {Tyler=330.0, Clark=363.0, Tanner=399.3, Lashanda=439.22998, Wilton=483.15298}
 Ranking {1}: {Jeanmarie=110.0, Astrid=121.0, Ina=133.1, Bertram=146.41, Garrett=161.05101}
 The winner of the auction {5} is Wilton and made an offer of 483.15298€
 The seller of the english bidding has been notified.
 The english auction {5} is closed
 Thanks for participating.
 Ranking {4}: {Stevie=330.0, Rudolph=363.0, Laurinda=399.3, Stan=439.22998, Debbi=483.15298, Devin=531.46826}
 The winner of the auction {1} is Garrett and made an offer of 161.05101€
 The winner of the auction {4} is Devin and made an offer of 531.46826€
 The winner of the auction {2} is Latina and made an offer of 5785.0933€
 The seller of the english bidding has been notified.
 The english auction {3} is closed
 Thanks for participating.
 The seller of the english bidding has been notified.
 The seller of the english bidding has been notified.
 The seller of the english bidding has been notified.
 The english auction {1} is closed
 Thanks for participating.
 The english auction {4} is closed
 Thanks for participating.
 The english auction {2} is closed
 Thanks for participating.
 finalising...
 shutting down...
 ending...

Conclusion

Pour conclure, la partie développement logiciel du projet comporte le protocole de deux enchères attendues, l'enchère à l'anglaise et l'enchère à l'offre scellée.

Une première implémentation a été réalisée de manière synchrone avec l'utilisation d'une architecture à base de composants. Ensuite, le projet est passé à une implantation asynchrone des protocoles avec une intégration de la gestion du parallélisme ainsi que de la concurrence pour permettre au commissaire-priseur de diriger plusieurs enchères à la fois. Pour les dernières étapes du projet, nous y avons intégré l'utilisation de greffons accordant la possibilité aux composants d'adopter plusieurs rôles, ainsi que les répartitions des composants dans des JVM différents.

Afin de réaliser des scénarios de tests, nous avons choisi d'intégrer un modèle de décision en lien avec les comportements des acheteurs potentiels au cours d'une enchère.

Notre projet pourrait bien évidemment être amélioré de plusieurs façons notamment réaliser un affichage graphique du déroulement de chaque enchère en cours dans une fenêtre séparée ou bien encore compléter le projet avec d'autres protocoles d'enchères comme l'enchère à la hollandaise ou bien encore l'enchère à la japonaise.

Pour rendre les scénarios de tests encore plus intéressants, nous aurions également pu nous concentrer davantage sur le modèle décisionnel en incluant le système de recommandation de bandit souvent utilisé dans le domaine de l'intelligence artificielle.

Il aurait été intéressant de tester la partie asynchrone sur des états de l'enchère. En d'autres termes, nous aurions pu lancer chaque état de l'automate d'enchère sur un thread au lieu de dérouler le protocole intégralement sur un thread.

Ce projet nous a permis de nous confronter à diverses thématiques du monde du développement logiciel tel que les écueils de la concurrence et l'asynchronie, de faire la différence entre des systèmes répartis et multi threadé ainsi que leurs enjeux. Ce qui nous a permis d'en ressortir plus riche en termes d'apprentissage et d'expérience.

Références

- [1] Anwar Sajid, Robert McMillan, et Mingli Zheng. « Bidding Behavior in Competing Auctions: Evidence from EBay ». *European Economic Review* 50, n° 2 (1 février 2006): 307-22. <https://doi.org/10.1016/j.eurocorev.2004.10.007>.

- [2] Valentin Robu, Han Noot, Han La Poutre, et Willem-Jan van Schijndel. « A Multi-Agent Platform for Auction-Based Allocation of Loads in Transportation Logistics ». *Expert Systems with Applications* 38, no 4 (1 avril 2011): 3483-91. <https://doi.org/10.1016/j.eswa.2010.08.136>.

- [3] Noriega Pablo, et Carles Sierra. « Auctions and Multi-Agent Systems ». In *Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet*, édité par Matthias Klusch, 153-75. Berlin, Heidelberg: Springer, 1999. https://doi.org/10.1007/978-3-642-60018-0_9.