




Mercado De Leilões

Grupo 29:

- Carlos Freitas - 201504749
 - Luis Martins - 201503344
 - João Conde - 201503256
- 

Descrição do problema



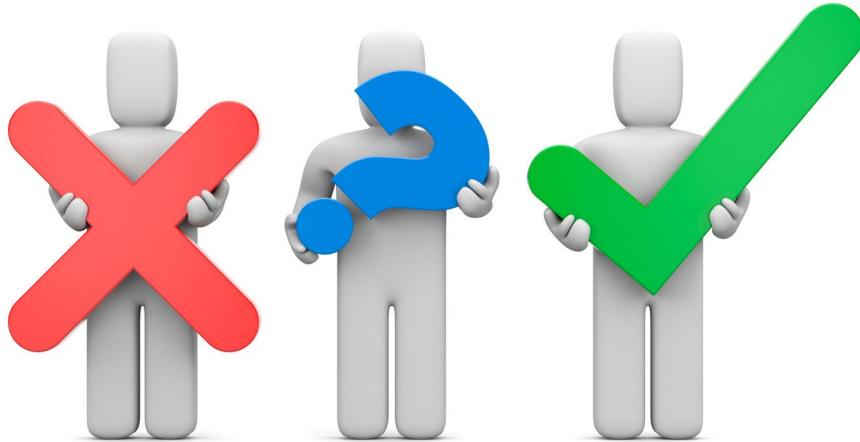
No cenário em questão, um ou mais compradores possuem uma lista de produtos que pretendem comprar. Simultaneamente, vários vendedores tentam leiloar os seus produtos.

Cada vendedor fixa um preço mínimo para cada produto que vende, sendo este o valor inicial do produto em leilão (English auction). O mesmo produto, oferecido por diferentes vendedores, pode ter preços diferentes.

Cada comprador estará disposto a gastar um valor máximo por cada produto. No entanto este valor pode reduzir caso o vendedor tenha má reputação (0-1).

Um comprador que ganhe 2 ou mais leilões do mesmo produto pode escolher qual prefere, conforme o rating dos vendedores que licitaram os produtos ganhos

Descrição do problema



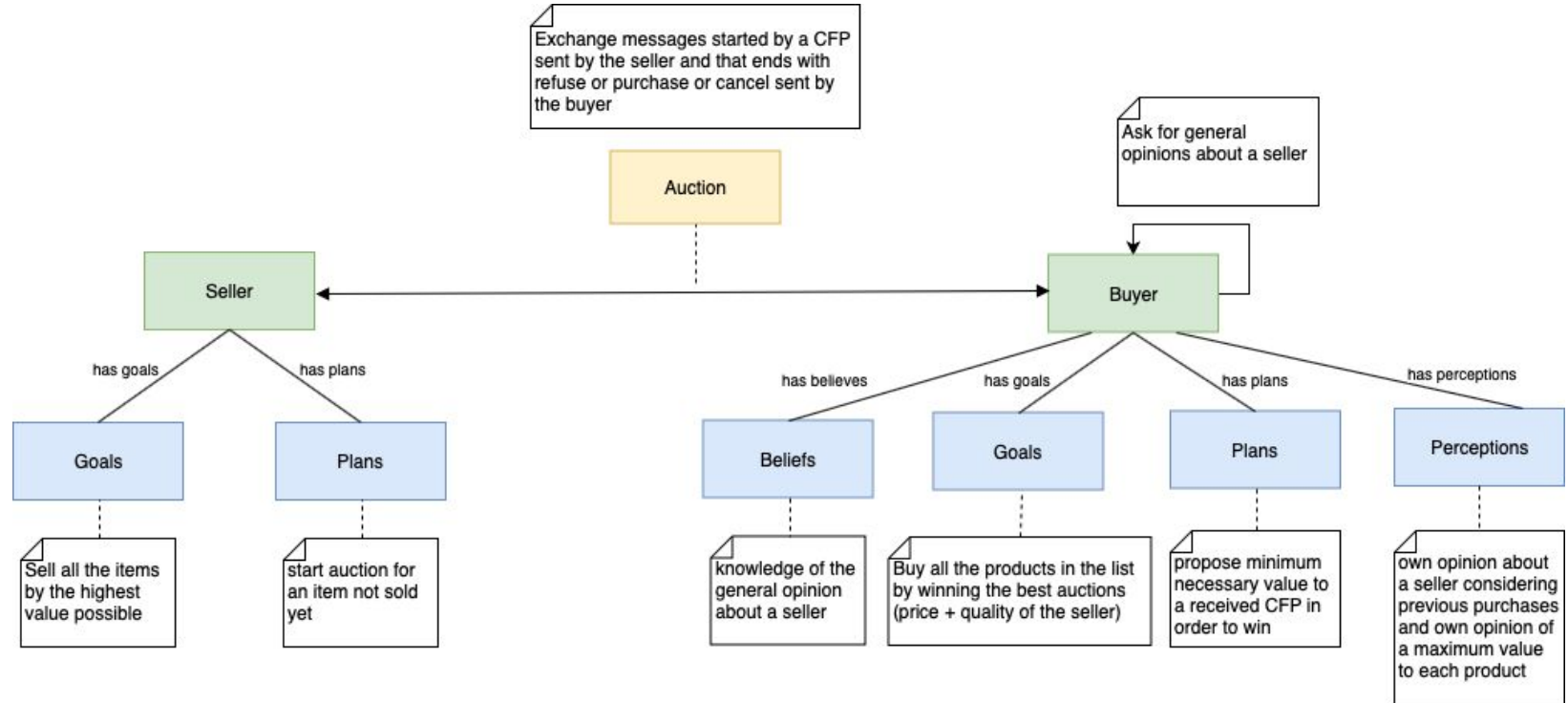
O rating de um vendedor é dado pelas opiniões dos compradores do sistema que já lhe compraram algum produto.

A opinião de cada comprador do sistema sobre os vendedores aos quais já comprou algum produto é calculada com base no tempo prometido de entrega dos produtos comprados e no tempo realmente verificado.

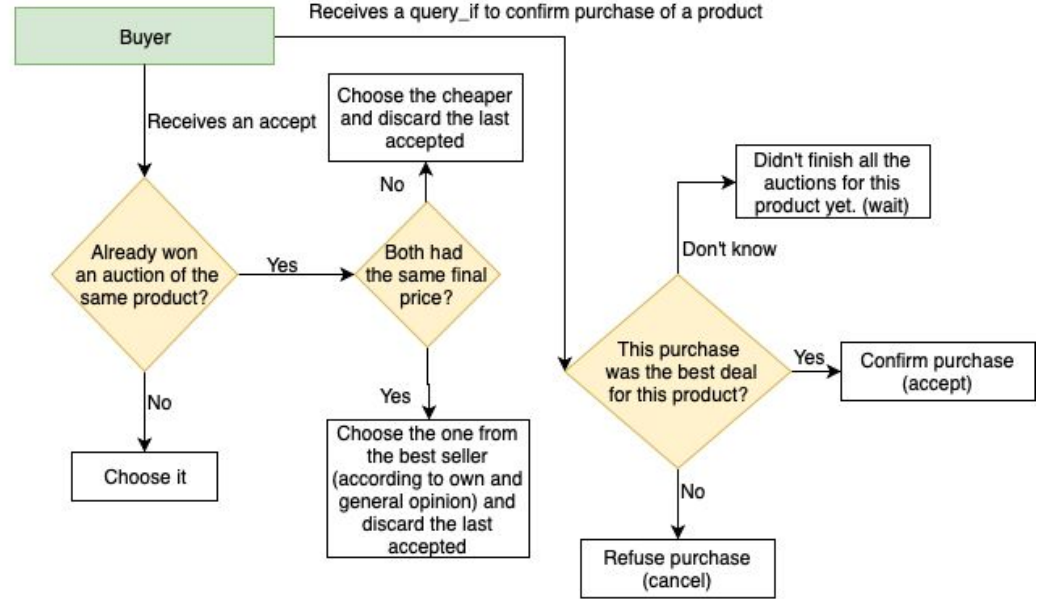
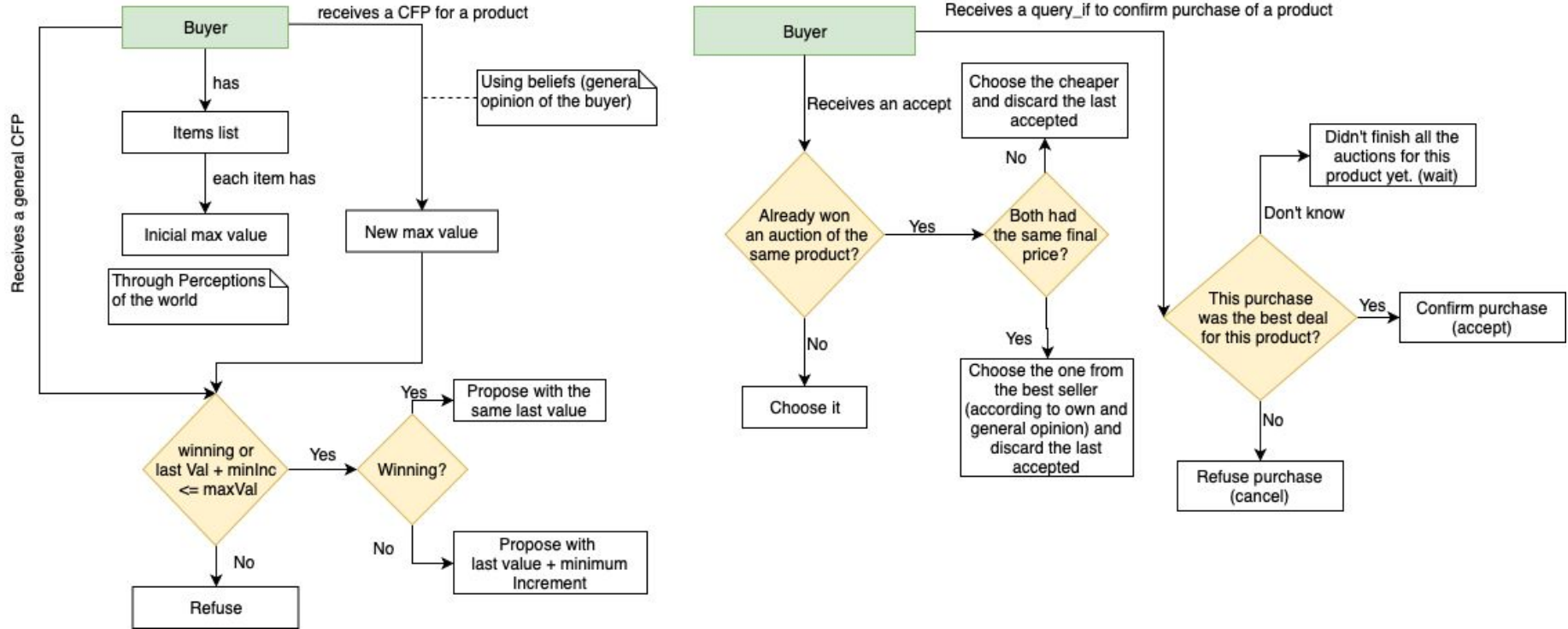
Caso não exista nenhuma transação prévia para o vendedor em questão, a sua reputação será 0.8.

Os agentes mantêm-se no sistema até conseguir comprar/vender todos os produtos que pretenderem ou até atingirem o tempo limite determinado

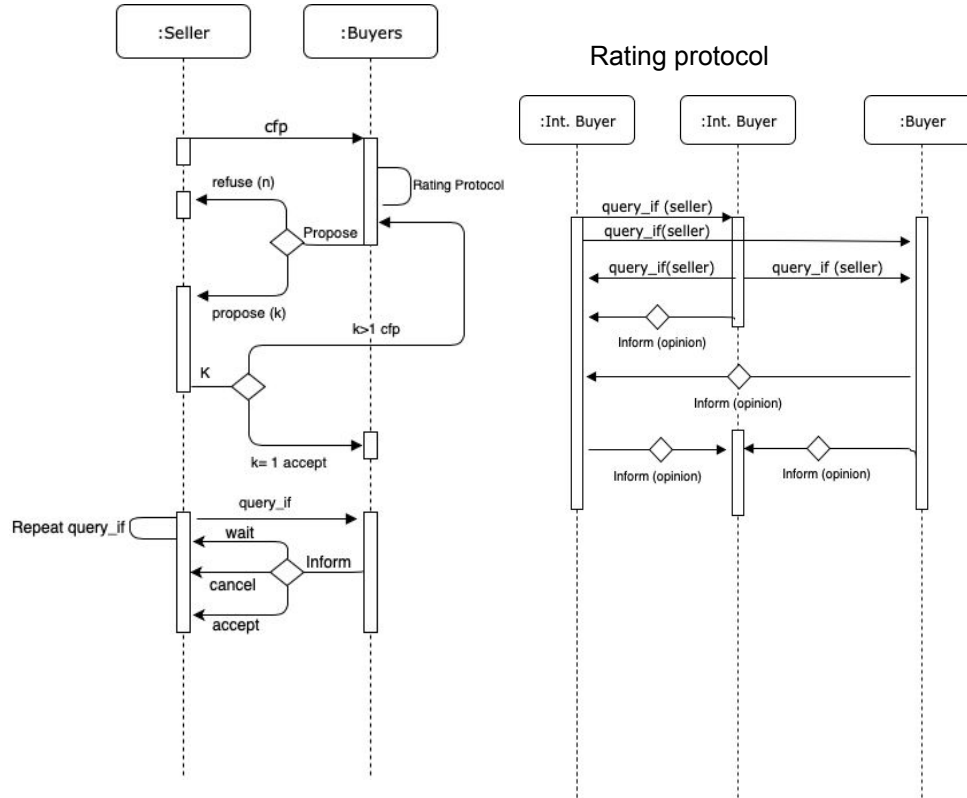
Esquema global agentes e como encaixam



Arquitetura agentes e estratégias utilizadas



Interação e protocolos comunicação



A comunicação relativa a cada auction é feita com recurso ao protocolo *IteratedContractNet*

O seller envia CFP para todos os buyers que estão interessados no produto para que estes proponham um valor.

Cada buyer que recebe um primeiro CFP de um leilão executa o rating protocol na função de interested Buyer (Int.Buyer)

As opiniões que um buyer envia como resposta à mensagem *query_if (seller)* podem ser:

- NULL: caso não conheça o seller
- [0-1]: caso já tenha opinião formada sobre o buyer (já lhe tenha comprado alguma coisa)

Utilização de páginas amarelas



Para que o seller saiba quais os buyers que estão interessados em cada produto que vende, é utilizado o agente DF (páginas amarelas).

Quando um buyer entra no sistema envia para o DF quais os produtos que tem interesse em comprar na forma de ServiceDescriptions.

Um seller quando pretende iniciar um leilão de um certo produto consulta as páginas amarelas por forma a obter uma lista de potenciais interessados e para os quais envia as mensagens de CFP de início de leilão.

Quando um buyer compra um produto, remove este mesmo produto da sua "lista" nas páginas amarelas de forma a não receber propostas para um leilão que já não lhe interessa.

Experiências

Os vendedores e os compradores que irão aparecer no mercado, são retirados a partir de ficheiros XML. Assim, é possível realizar e testar facilmente vários exemplos a partir da alteração das características dos vários indivíduos ou da adição de novos.

```
<market>
  <buyer name="Alberto">
    <item maxvalue="15.0">batatas</item>
  </buyer>
  <buyer name="Bruno">
    <item maxvalue="12.5">batatas</item>
    <item maxvalue="17">bananas</item>
  </buyer>
  <buyer name="Carlos">
    <item maxvalue="25">batatas</item>
    <item maxvalue="15">bananas</item>
  </buyer>
</market>
```

```
<market>
  <seller name="Tomas" shipmentDelay="0">
    <bid>
      <item price="1">batatas</item>
      <delivery>3</delivery>
      <increase>0.6</increase>
    </bid>
  </seller>

  <seller name="Rui" shipmentDelay="0">
    <bid>
      <item price="1">batatas</item>
      <delivery>10</delivery>
      <increase>0.8</increase>
    </bid>
  </seller>
</market>
```


Análise

Por cada vez que o programa é corrido, é gerado automaticamente ficheiros de logs que demonstram facilmente todo o processo ocorrido, ou seja, todas as interações realizadas entre os vários agentes e consequentemente os estados dos vários leilões que ocorrem simultaneamente, com estes logs é possível retirar várias conclusões sobre cada um dos agentes.

- Buyer

- Percentagem de produtos que conseguiu comprar
- Quanto leilões concorreu para obter um produto
- Percentagem de leilões que ganhou
- Para cada produto que ganhou, quanto pagou a mais em relação ao preço base
- Média que gastou a mais do valor base dos itens

- Seller

- Quantas vezes uma venda foi cancelada
- Para cada item em leilão, quantas vezes foi proposto um valor
- Para cada item vendido, o rendimento obtido
- Rendimento total
- Rendimento médio por item

Conclusões

Neste trabalho foi implementado um mercado de leilões no formato *English Auction*. Uma vez que se trata de um sistema multi-agente, foram então feitas algumas alterações ao modelo clássico, nomeadamente relativo ao incremento de valor que cada participante faz quando lícita, ou seja, tradicionalmente o mínimo incremento por licitação deveria ir aumentando, o que não acontece neste caso.


O grupo considera ter cumprido todos os objetivos propostos, no entanto seria interessante para trabalho futuro implementar mais algumas funcionalidade nomeadamente adicionar personalidades nos compradores de forma a que não subissem sempre o mesmo valor em cada leilão, uns arriscam mais outros não. Outra melhoria possível seria tentar melhorar a eficácia de venda de um vendedor fazendo-o propor a venda de um produto ao segundo melhor licitador caso o vencedor da licitação recuse a compra do mesmo.

Visto tratar-se de um sistema distribuído sem noção de tempo, este projeto requer algumas verificações e cuidados de forma a que o sistema não dê nenhum erro.



Parte 2

Informação adicional



Exemplos detalhados de execução

- Leilão simples com dois compradores o Miguel e o Carlos que disputam por um mesmo produto.
 - O vendedor Conde inicia um leilão de meias e manda uma proposta aos dois compradores interessados.
 - Miguel e Carlos recebem a proposta, antes de aceitar, perguntam um ao outro qual a opinião que têm do Conde, como é o primeiro leilão ainda não há opinião, por isso cada um dos compradores assumem o rating predefinido de 0.8.
 - O leilão inicia com um valor mínimo de 10. O Miguel e o Carlos disputam pelo item propondo novos valores para o item sempre com um incremento de 0.5 valores.
 - O Carlos está disposto a gastar no máximo 11.6 ($0.8 * 14.5_{(a)}$) e o Miguel 12 ($0.8 * 15.0_{(a)}$)
 - Eventualmente o Miguel propõe pagar 12 pelas meias e conseqüentemente o Carlos desiste do leilão.
 - Miguel ganha.
 - Conde tenta confirmar a compra com o Miguel.
 - Miguel aceita a compra. Verifica que já comprou tudo o que desejava e sai do mercado.
 - Conde não tem mais nada a vender e sai do mercado
 - Carlos ainda continua interessado em meias e fica até 25 segundos à espera de uma nova proposta para adquirir meias.
 - Passado 25 segundos o Carlos desiste e sai do mercado.

Exemplos detalhados de execução

- Mercado com dois vendedores, o Tomás e o Rui. O leilão de um é cancelado,e eventualmente reiniciado.
 - O Tomás quer vender batatas e o Rui quer vender batatas e bananas.
 - Os três compradores no mercado , o Alberto, o Bruno e o Carlos estão interessados em batatas.
 - O Tomás e o Rui mandem propostas aos compradores para iniciar o leilão de batatas.
 - Os compradores competem entre si nos dois leilões,simultaneamente.
 - O Carlos como está disposto a gastar mais pelas batatas, ganha os dois leilões.
 - O Carlos avalia dos dois leilões que ganhou qual o melhor negócio. Cancela a compra ao Rui e compra as batatas ao Tomás.
 - O Rui decide tentar vender o próximo produto que tem,logo inicia um leilão para as bananas.
 - O Bruno e o Carlos competem pelas bananas.
 - Bruno ganha o leilão e compra as bananas ao Rui.
 - Rui repara que ainda tem as batatas por vender, então tenta novamente iniciar o leilão para as batatas.
 - O Bruno e o Alberto ainda interessados em obter batatas, competem entre si no leilão do Rui.
 - Alberto ganha e compra as batatas ao Rui.
 - Rui observa que vendeu todos os seus itens e sai do mercado.

Outros exemplos de execução

- Um vendedor é mentiroso e diz que consegue entregar os itens que leiloa em muito menos tempo do que realmente demora. Consequentemente por cada leilão que faz o vendedor fica com má fama devido ao ratings baixos que são dados pelos compradores
- Todos os conseguem obter todos os itens das suas listas de desejos.

Classes implementadas

Estruturamos o projeto em dois packages, **utilities** e **main**. Em cada um definimos os seguintes ficheiros .java e respetivas classes:

No package **utilities**:

1. **Utils** - métodos utilitários estáticos, definições de protocolos e respetivas mensagens
2. **MarketLogger** - gestão dos ficheiros com resultados de execução (.log files)
3. **BuyerStatistics** - recolha de dados de cada BuyerAgent e cálculo de estatísticas finais
4. **SellerStatistics** - recolha de dados de cada SellerAgent e cálculo de estatísticas finais

No package **main**:

1. **Market** - execução do JADE, criação do main container e agentes
2. **SellerAgent** - agente vendedor com diferentes itens para vender
3. **BuyerAgent** - agente comprador com diferentes itens para comprar
4. **Bid** - estrutura representante de uma aposta
5. **Purchase** - estrutura representante de uma compra
6. **RatingInfo** - estrutura auxiliar na recolha de opiniões de outros BuyerAgent 's

SellerAgent

A classe **SellerAgent** estende a classe `jade.core.Agent` e representa um vendedor.

Um vendedor procura vender os seus itens e, enquanto não os vender todos, executa um behaviour sequencial. Este é composto por dois behaviours, um focado na procura dos compradores interessados no produto e o outro iniciando a negociação com todos os interessados (**FIPACONTRACTNetInitiator**).

O leilão prossegue enquanto existir mais do que um comprador, a licitar mais e mais alto. Depois da desistência de todos menos um, o comprador que sobrar é considerado vencedor.

SellerAgent

O vendedor comunica com o comprador a informá-lo da vitória no leilão através do seu **BidKeeper** behaviour. O vendedor ficará à espera de uma resposta do comprador pois este pode desistir do leilão mesmo após vencê-lo (por exemplo se ganhar um outro leilão para o mesmo produto, mas mais barato ou de um vendedor com melhor reputação).

Se o comprador quiser o produto, o vendedor considera-o vendido e passa à venda do próximo item, se tiver.

Se o comprador cancelar a compra, o vendedor re-inicia o leilão do produto.

BuyerAgent

A classe **BuyerAgent** estende a classe `jade.core.Agent` e representa um comprador.

Um comprador está à escuta de eventuais leilões que lhe interessem, isto é, se um leilão de um produto por ele desejado começa. Para o efeito, possui um **CFP dispatcher** que estende a classe **SSResponderDispatcher** responsável por criar um **FIPAlteredContractedNet** (estende a classe de `jade.SSIteratedContractNetResponder`) por leilão. Isto permite que um mesmo comprador licite em vários leilões simultaneamente.

Um comprador possui ainda um **QUERY_IF dispatcher** para responder à tentativa de confirmação de compra do produto pelo vendedor do leilão que ganhou. A resposta ao mesmo pode ser de **WAIT**, **CANCEL** ou **PURCHASE** correspondendo, respetivamente, à necessidade do vendedor esperar, reconhecer como cancelada a compra ou de confirmar a compra.

BuyerAgent

Antes de licitar num novo leilão, o comprador procura averiguar a reputação (ou rating) do vendedor. Para isso, pede a opinião de outros compradores que já tenham transacionado com este vendedor e combina com a sua própria percepção do vendedor em questão.

Este rating combinado afeta a quantidade de dinheiro que o comprador está disposto a dar por este produto a este vendedor.

Para tal, o agente comprador pergunta no primeiro CFP do leilão pela opinião alheia e mantém registo de tal.