

Trabalho Prático 2: Leilão Japonês.

George Souza Oliveira*

Daniel Domingos Adriano†

2019, v-1.0

Resumo

O leilão japonês é um leilão semelhante ao leilão inglês, onde o preço do produto é ajustado pelos compradores interessados, com o propósito de encontrar o comprador vencedor, que é o dono do maior lance. Há duas classes de integrantes que compõem o leilão japonês: o leiloeiro e os apostadores. O leiloeiro é aquele que anuncia um produto e o seu valor; e os apostadores (ou compradores interessados) são os integrantes restantes que aceitam o lance do produto associado. Este trabalho apresenta três versões de implementação do leilão japonês por meio de linguagens de programação multi-agentes: VERSÃO A, que utiliza a linguagem **Jason**; a VERSÃO AE, que utiliza **Jason** e **CArtAgO** e a VERSÃO AEO, que emprega **Jason**, **CArtAgO** e **Moise**. Os experimentos mostraram maior velocidade de execução da VERSÃO AEO, principalmente pela inexistência de *delay* da troca de mensagens e das constantes interações com o artefato.

1 Introdução

O leilão japonês (1), também conhecido como leilão de botão, é um leilão semelhante ao leilão inglês (2), onde o preço do produto é ajustado pelos compradores interessados, com o propósito de encontrar o comprador vencedor, que é o dono do maior lance.

A grande diferença do leilão japonês em relação aos leilões tradicionais está no fato de que os compradores *devem* aceitar a oferta de cada rodada de lances para continuarem ativos no leilão. Os desinteressados são obrigados a deixar o leilão e são impossibilitados de retornar à disputa.

Uma vantagem significativa é a simplicidade para participar das licitações, uma vez que basta ouvir os lances e declarar o interesse. Por outro lado, uma grande desvantagem é que, mesmo que os compradores tenham a intenção de dar maiores lances em outras rodadas, ainda é possível ficar fora da disputa caso não aceitem a oferta atual.

Voltado ao leilão japonês, este relatório apresenta as descrições de implementação e análise comparativa de três versões de solução do mesmo, empregando sistemas multi-agentes (SMA), segundo especificações definidas em aula. As versões são:

- **Versão A:** Somente os agentes são programados (empregando Jason);

*geo.soliveira@gmail.com

†daniel.dadriano@gmail.com

- **Versão AE:** O ambiente pode ser instrumentado para ajudar o leilão (utilizando CArtAgO);
- **Versão AEO:** Uma organização é estabelecida para o leilão (com Moise).

Este relatório é dividido nas seguintes seções: A Seção 2 apresenta a contextualização do problema de leilão japonês; a Seção 3 apresenta as especificações das versões desenvolvidas; a Seção 4 apresenta uma análise comparativa e a Seção 5 finaliza com as conclusões.

2 Contextualização do Problema

Há duas classes de integrantes que compõem o leilão japonês: o leiloeiro e os apostadores. O leiloeiro é aquele que anuncia um produto e o seu valor; e os apostadores (ou compradores interessados) são os integrantes restantes que aceitam o lance do produto associado.

Dado um comprador c qualquer que *acredita* que o preço do produto p anunciado seja v_c . Enquanto o valor *real* anunciado pelo leiloeiro no tempo t é e_t , o comprador c permanecerá na arena de leilão enquanto $e_t < v_c$. Quando $e_t \geq v_c$, então c sairá da arena e não será mais capaz de adquirir e pagar pelo produto p . Isso classifica o leilão japonês como um leilão *honesto*, pois a ação é tomada de acordo com o verdadeiro valor do produto, isto é, e_t , e não do valor que os compradores acham que p tem (ou seja, v_c).

Como resultado, quando todos os compradores adotam tal estratégia, então o comprador vencedor será aquele com a avaliação mais alta, ou seja, o último comprador que sobrar na arena.

O leilão japonês prossegue da seguinte forma:

1. O leiloeiro exibe o preço inicial do produto. Esse preço aumenta continuamente ou em pequenas etapas discretas;
2. Todos os compradores interessados em comprar o produto entram na arena do leilão;
3. Cada comprador pode sair da arena em qualquer período, mas não pode voltar a entrar nela;
4. Quando um único comprador permanece na arena, o leilão é interrompido. O comprador restante (ou comprador vencedor) ganha o produto e paga o preço dele;
5. Mais de um comprador pode sair da arena ao mesmo tempo. Portanto, pode ocorrer (e é permitido) que não se encontre um comprador vencedor;

3 Soluções

Esta seção apresenta os detalhes de implementação das versões propostas, abordadas nas próximas seções.

3.1 Versão A

A VERSÃO A é a versão mais simples e foi implementada em **jason**. O leiloeiro é o agente **bob** e os participantes são os agentes **alice**, **maria**, **theo**, **john** e **jack**. O leiloeiro está leiloando um cavalo e inicia informando os demais agentes (os participantes), o preço do animal.

Cada agente participante possui um valor de compra associado a ele, calculado de forma randômica. Quando recebem a notificação de preço enviada pelo leiloeiro, os participantes comparam o preço do cavalo com o valor de compra. Se o valor de compra é maior, então o participante notifica o leiloeiro que, por sua vez, o registrará como ativo e avisará com a mensagem “*N is a participant*”, onde N é o nome do agente que aceitou a proposta. Caso contrário, o participante notifica o leiloeiro com a mensagem “*The horse is to expensive for me*” e se retira da disputa.

A cada um segundo, o leiloeiro aumenta o preço do cavalo, por um incremento aleatório, e notifica os participantes, os quais farão a avaliação do novo preço e notificarão o leiloeiro. Se um agente participante outrora ativo enviar a notificação “*The horse is to expensive for me*”, então o leiloeiro o removerá da lista de participantes ativos.

Vence o único agente que sobrar na lista de participantes ativos. A notificação é dada pela mensagem “*The winner is W*”, onde W é o nome do agente vencedor.

3.2 Versão AE

Na VERSÃO AE, o leiloeiro ainda é o agente **bob** e os participantes são os agentes **alice**, **maria**, **jósé**, **pedro**, **joão** e **jack**. Não é especificado um produto específico para o leilão como na VERSÃO A. Os preços e intenções de participação não são mais anunciados pela troca de mensagens, mas empregando o artefato **auctioneer_arena**, conforme mostrado na Figura 1. Esse artefato foi desenvolvido em **Java**, como uma classe nomeada **ArenaArtifact**, cujos métodos e atributos são:

1. **product_value**: Variável que armazena o preço atual do produto anunciado;
2. **participants_size**: Variável iniciada com zero, mas que incrementa na medida em que os participantes interessados entram na arena;
3. **initiateArena**: Método que inicia a arena, armazenando em **product_value** o preço inicial do produto. Também emite o sinal **initiate_arena** para os participantes;
4. **updateValue**: Método que atualiza o preço armazenado em **product_value**. Também emite o sinal **value_updated** para os participantes;
5. **winner**: Método que devolve o participante vencedor da disputa. Também emite o sinal **winner** para os participantes;
6. **joinArena**: Método que adiciona o participante interessado na disputa na arena e incrementa **participants_size**;
7. **exitArena**: Método que remove o participante da arena e decrementa **participants_size**.

No início da missão, o leiloeiro **bob** define randomicamente o preço para um produto genérico e invoca **initiateArena**. Depois, a sua única missão é atualizar o preço do produto

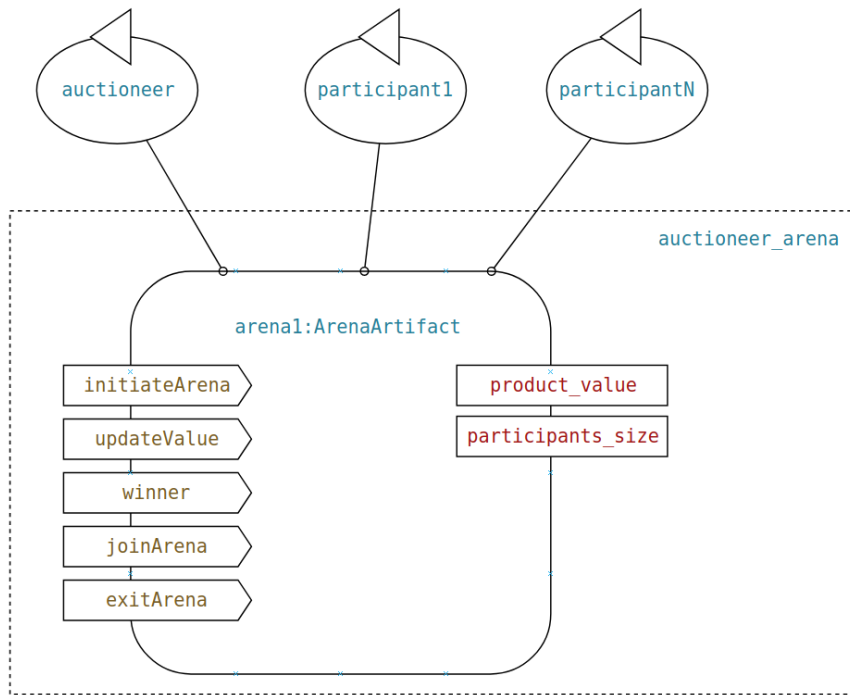


Figura 1 – Especificação da VERSÃO AE

a cada segundo. Nesse processo, três casos podem ocorrer: (1) `participants_size` é maior que 1, o que obriga bob a atualizar `product_value`, invocando `updateValue`; (2) `participants_size` é igual a 1, o que obriga bob a anunciar o vencedor da disputa, por meio do método `winner`; (3) `participants_size` é igual a 0, o que obriga bob a encerrar o leilão sem anunciar um vencedor.

Por sua vez, cada participante adquire um valor de compra (também randomicamente). Depois disso, eles aguardam os sinais emitidos pelos artefatos e agem conforme cada recebimento, que podem ser:

- Se o sinal é `initiate_arena`, então o participante compara o preço do produto com o seu valor de compra. Se o valor de compra for maior, então ele invoca o método `joinArena`. Este é o único momento em que um participante pode entrar na arena.
- Se o sinal é `value_updated`, então pode ocorrer dois casos: se o valor de compra for maior que o preço do produto, então o participante responde com “*Still a good price*” e permanece na arena; se o valor de compra for menor, então o participante responde com “*The product is too expensive for me*” e sai da arena, invocando `exitArena`.
- Se o sinal é `winner`, então o agente verifica se o vencedor foi ele mesmo. Se sim, ele responde com “*Bought!!*” ao mesmo tempo que o leiloeiro avisa com “*The winner is W*” (onde W é o nome do agente vencedor).

3.3 Versão AEO

Nesta última versão, permanece os mesmos agentes que em AE e o agente bob ainda é o leiloeiro. A única iniciativa é por parte dos agentes participantes, os quais definem

um valor de compra para si. Nesta versão existe uma organização, chamada ARENA, com suas especificações e normas bem definidas e que tem a função de outorgar as missões aos agentes (Figura 2). As especificações estrutural, funcional e normativa são apresentadas nas Tabelas 1, 2 e 3, respectivamente.

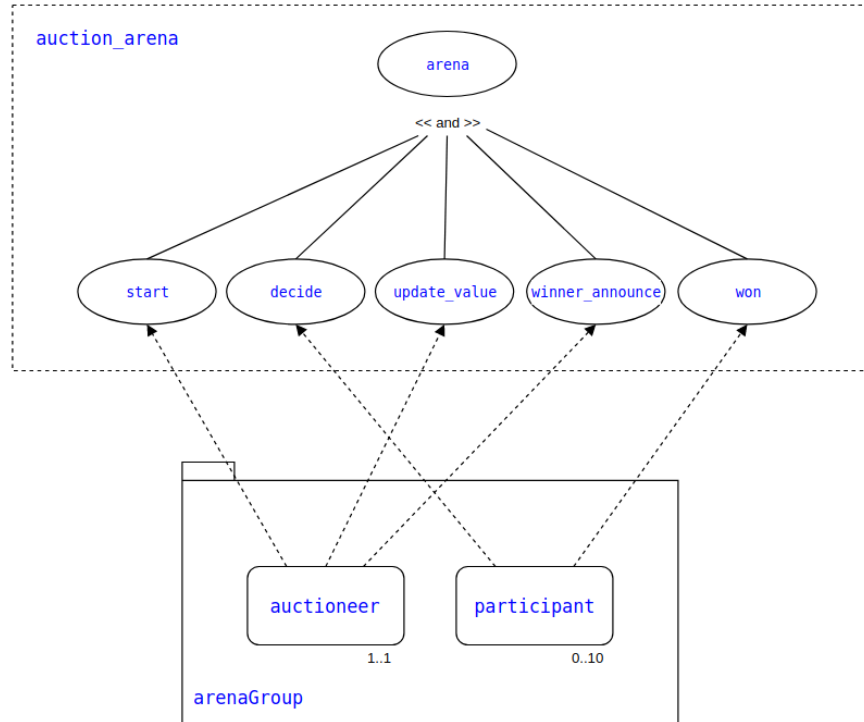


Figura 2 – Especificação da VERSÃO AEO

Tabela 1 – Especificação estrutural da organização ARENA. É exigido, no máximo, 10 participantes e 1 só leiloeiro.

Agentes	Quantidade mínima	Quantidade máxima
Leiloeiro	1	1
Participantes	0	10

Tabela 2 – Especificação funcional da organização ARENA. As missões são *vender* e *comprar*.

Missão	Objetivos	Especificação no Código	Deadline
Vender	Iniciar	start	Não há
	Atualizar valor	update_value	60 segundos
	Anunciar vencedor	winner_announce	Não há
Comprar	Decidir	decide	10 segundos
	Anunciar que é o vencedor	won	Não há

Referente as Tabelas 2 e 3, a ARENA define duas missões aos agentes: vender e comprar. A venda está associada ao agente **bob**, que tem o papel de leiloeiro, e ele cumprirá a sua missão se obter êxito em iniciar (**start**), atualizar o valor do produto (**update_value**) e anunciar o vencedor (**winner_announce**). A compra está associada aos

Tabela 3 – Especificação normativa da organização ARENA. O leiloeiro tem o papel de participar da venda e os participantes têm o papel de participar da compra. Isso não implica que sempre haverá um vencedor.

Tipo da norma	Papel	Missão	Descrição
Obrigaç�o	Leiloeiro	Vender	O leiloeiro est� obrigado a participar da venda
Obrigaç�o	Participantes	Comprar	Os participantes est�o obrigados a participar da compra

demais agentes, os quais possuem papel de participantes do leil o. Eles cumprir o a sua miss o decidindo (ou n o) pela perman ncia na disputa (**decide**) e, no caso do participante vencedor, anunciando a vit ria (**won**).

Os objetivos vinculados aos pap is dos agentes associados   ARENA s o:

- **Iniciar** (especifica  o no c digo: **start**): Inicializa o leiloeiro;
- **Atualizar valor** (especifica  o no c digo: **update_value**): O leiloeiro atualiza o valor do produto;
- **Anunciar vencedor** (especifica  o no c digo: **winner_announce**): O leiloeiro anuncia o vencedor da aposta;
- **Decidir** (especifica  o no c digo: **decide**): Os participantes decidem se continuam no leil o;
- **Anunciar que   o vencedor** (especifica  o no c digo: **won**): O participante vencedor anuncia sua vit ria.

A VERS O AEO funciona como a VERS O AE. No programa do agente que faz o papel de leiloeiro, os predicados (ou regras) s o acionados pela organiza  o, conforme ordem de preced ncia. Nos participantes, o predicado **start** (n o confundir com o objetivo **start** definido na organiza  o)   acionado na inicializa  o e o predicado **value_updated**   acionado pelo **signal** do artefato, como na VERS O AE. Os predicados/regras **decide** e **won** s o acionados pela organiza  o, cujas fun  es est o descritas no par grafo supracitado.

4 Experimentos

Esta se  o apresenta um experimento no sentido de medir o tempo de execu  o das vers es implementadas. Embora o valor inicial do produto e os valores de compra dos agentes participantes sejam, por padr o, definidos randomicamente, para este experimento eles foram fixados. Os par metros definidos para todas as vers es foram:

1. O valor inicial do produto   2;
2. O valor inicial   atualizado a cada 1 segundo. Sendo V o valor inicial, a f rmula para atualizar o valor de V   $V = V + (\text{math.random} \times V)$;
3. Os agentes participantes possuem valores de compra fixos definidos como na Tabela 4;

4. Cada versão foi executada por 3 vezes.

Tabela 4 – Valores de compra dos agentes na três versões (definidos para o experimento).

Agente	Valor de compra
Alice	30
Maria	25
Theo	35
John	40
Jack	38

A Tabela 5 mostra o tempo de por execução e o tempo médio para cada versão. Em média, a VERSÃO AEO obteve melhor desempenho por conta do gerenciamento própria da organização. A VERSÃO AE não foi equiparável à VERSÃO AEO por conta do *delay* que existe na manipulação constante do artefato por todos os agentes. Em AEO um artefato também é utilizado, mas em menor proporção, principalmente no uso da diretiva **signal** (utilizada somente para acionar a regra **value_updated**). O tempo maior de execução da VERSÃO A pode ser explicado pelo atraso inerente do envio e recebimento de mensagens entre os agentes.

5 Conclusões

Este trabalho apresentou os detalhes de implementação do leilão japonês por meio da aplicação de sistemas multi-agentes. Foram implementadas três versões: VERSÃO A, que utiliza a linguagem **Jason**; a VERSÃO AE, que utiliza **Jason** e **CARTAgO** e a VERSÃO AEO, que emprega **Jason**, **CARTAgO** e **Moise**. Os experimentos mostraram maior velocidade de execução da VERSÃO AEO, principalmente pela inexistência de *delay* da troca de mensagens e das constantes interações com o artefato.

Este trabalho finaliza com as experiências adquiridas pelos autores ao desenvolver cada versão:

- A VERSÃO A foi a implementação mais complicada, visto que todo o código teve que ser implementado nos agentes. A complexidade foi ainda maior por conta da necessidade de tratar as mensagens enviadas e recebidas. A impressão é que o uso de mensagens torna o programa menos flexível, o que dificulta uma abordagem mais complexa.
- Com o artefato, a VERSÃO AE se tornou mais fácil, principalmente pela ausência de mensagens. A capacidade de os agentes interceptarem as alterações no artefato (pela diretiva **signal**) foi outro ponto que contribuiu para essa avaliação positiva.
- Pouca coisa mudou entre a VERSÃO AE e a VERSÃO AEO. A impressão é que a última versão pareceu mais dinâmica tendo em vista a possibilidade de definir a sequência das tarefas sem se preocupar com qualquer implementação adicional, seja nos agentes ou seja nos artefatos.

Referências

- 1 SCHINDLER, J. Late bidding on the internet. 04 2003. Citado na página 1.
- 2 MCAFEE, R. P.; MCMILLAN, J. Auctions and bidding. *Journal of Economic Literature*, v. 25, n. 2, p. 699–738, 1987. Disponível em: <<https://www.jstor.org/stable/2726107>>. Citado na página 1.