Negociação Automatizada em Comércio Móvel Utilizando Agentes Móveis

Fernando M. Matos Edmundo R. M. Madeira

IC - Instituto de Computação
UNICAMP - Universidade Estadual de Campinas
13083-970 Campinas, SP
{fernando.matos,edmundo}@ic.unicamp.br

Resumo

A facilidade de aquisição e a queda nos preços dos aparelhos de telefones celulares e PDAs estão contribuindo para o aumento do número de pessoas que acessam a Internet através de dispositivos móveis e consequentemente de pessoas que realizam tansações através deles, o chamado *m-commerce* (*mobile commerce*). Este artigo propõe um protocolo de negociação automatizada entre dois participantes para comércio móvel, utilizando agentes móveis e levando em consideração a personalização dos dispositivos móveis através da utilização de perfis na negociação. A característica de agentes móveis de agirem de forma autônoma e independente é uma grande vantagem do seu uso em comércio móvel, uma vez que neste tipo de ambiente a conexão do usuário com a rede fixa é intermitente. Este protocolo baseia-se na especificação do OMG. A implementação do protótipo, que utiliza o ambiente de agentes móveis Grasshopper como infra-estrutura de comunicação, é comentada. Como exemplo de aplicação, a compra de passagens aéreas é apresentada.

1 Introdução

Com a enorme expansão que aconteceu com a Internet, cada vez mais pessoas têm acesso às facilidades e comodidades que a acompanham. Uma destas comodidades é o comércio eletrônico, no qual uma pessoa, sem precisar sair do conforto de sua casa, pode comprar quase todo tipo de produto. Dificilmente este "novo" conceito de comércio tomará o lugar do comércio convencional, pelo menos por enquanto. Mas é inegável o fato de que ele vem atraindo um número cada vez maior de consumidores [12].

Devido a isto, nada mais natural do que automatizar cada vez mais todo o processo de compra, o que vai desde a procura do produto, passando pela negociação, até a compra propriamente dita do produto. O trabalho [10] propõe um modelo de negociação automatizada. Este trabalho utiliza agentes móveis para realizar a negociação de preço de produtos de forma automatizada. A negociação é realizada enviando agentes de compra para os *hosts* dos vendedores. Uma vez lá eles negociam o preço do produto com agentes de venda e, depois, retornam o resultado para o agente de compra original para que o cliente possa se decidir pela compra.

Uma variante do comércio eletrônico é o chamado comércio móvel (*m-commerce*). Devido à grande quantidade de telefones celulares e PDAs existentes atualmente, é natural que as pessoas queiram realizar o maior número de atividades por eles. Estas atividades vão desde a consulta meteorológica até a compra de bens e serviços. Para tanto é necessário que se desenvolva mecanismos e técnicas de comunicação para o comércio móvel que atendam aos requisitos e limitações dos aparelhos. Estas limitações vão desde o tamanho reduzido da tela dos telefones celulares até a baixa largura de banda das redes sem fio [9].

Outra questão que se deve levar em consideração quando se trata de comércio móvel é o fato dos dispositivos móveis serem pessoais. Devido a isto é aconselhável, ou até necessário, a preocupação com o perfil do usuário. Elaborar uma maneira de conseguir realizar transações utilizando este perfil tornaria o processo de compra cada vez mais semelhante ao que é realizado na vida real, garantindo uma maior personalização da compra.

Por fim, um importante aspecto de comércio móvel diz respeito à segurança. Faz-se necessário o desenvolvimento de mecanismos que possam assegurar a confiabilidade, integridade e sigilo das informações. Porém, esta questão não é o foco deste trabalho. Contudo, nada impede que sejam aplicadas neste trabalho técnicas de segurança usuais, tanto em agentes móveis quanto em comércio móvel.

Este artigo propõe uma extensão do modelo proposto em [10] para comércio móvel. As extensões propostas se baseiam no pouco poder de processamento de aparelhos portáteis, na tela pequena dos aparelhos, na baixa largura de banda das redes sem fio e pricipalmente na utilização do perfil do usuário.

Ainda vale ressaltar que o uso de agentes móveis é especialmente interessante para este trabalho, uma vez que não é necessário, nem possível, que o usuário fique permanentemente conectado, já que o aparelho possui pouca autonomia devido ao uso de baterias.

O modelo de negociação do OMG [5] é utilizado como base para realizar a negociação pois, além de ser um padrão mundialmente aceito, [10] também se baseia neste modelo.

Na seção 2 serão apresentados os modelos do OMG e de [10]. A seção 3 apresenta o modelo proposto por este artigo. A seção 4 mostra alguns apectos de implementação do protótipo desenvolvido, e na seção 5 é apresentada a conclusão e alguns trabalhos futuros são apresentados.

2 Modelos de Negociação do OMG e de [10]

Este trabalho estende o modelo de negociação apresentado em [10], que propõe um protocolo de negociação entre dois participantes. O modelo de negociação é composto por agentes de compra e venda que negociam entre si de forma que possam conseguir o melhor acordo. A negociação de preço é baseada no modelo Kasbah [2]. Os catálogos do modelo são implementados em XML [11] para fornecer a interoperabilidade entre diferentes sistemas de agentes. [10] se baseia no modelo de negociação bilateral proposto pelo OMG (Figura 1), onde duas partes negociam com a finalidade de alcançar um acordo de benefício mútuo.

O modelo de negociação do OMG possui os estados *open* e *closed*. *Open* possui dois subestados: *requested* e *offered*, sendo que o último possui o subestado *proposed*. *Closed* possui os subestados *agreed*, *rejected* e *timeout*. A negociação pode ser inicializada em qualquer um dos três estados: proposto (*proposed*), requisitado (*requested*) ou ofertado (*offered*). *Offered* indica um estado em que o assunto da negociação não pode ser alterado. O estado *proposed* permite que o assunto da negociação possa ser alterado através da transição *request*, a qual leva o modelo ao estado *requested*. O estado *requested* permite a um participante transitar para o estado *offered* ou *proposed* através das transições *offer* ou *propose* respectivamente. Os estados *offered* e *proposed* denotam um estado de grandes possibilidades de que os participantes cheguem a um acordo mútuo. O estado *requested* indica um estado onde não há um compromisso entre os participantes. Uma negociação é considerada terminada quando um estado *closed* é alcançado.

A seguir é apresentada uma breve descrição das transições do modelo:

Transições

Request

É uma transição que pode ser aplicada no estado *proposed*. Ela permite uma alteração do assunto da negociação e a mudança do contexto através da transição do estado *proposed* para o estado *requested*.

Suggest

Esta transição é semanticamente equivalente à transição *request*, com a diferença que ela é usada no estado *requested*. Esta transição é utilizada como um mecanismo de busca na qual os dois participantes invocam sugestões um do outro, até que um deles esteja apto a alcançar um nível mais alto de compromisso, o qual é definido pelos estados *proposed* e *offered*.

Propose

É uma transição que ocorre do estado *requested* para o estado *proposed* e que indica um maior compromisso da parte que fez a proposta, já que ela pode ser aceita pela outra parte da negociação.

Offer

É uma transição do estado *requested* para o estado *offered*. Ela indica um acordo pela parte que fez a oferta, e restringe a possibilidade de uma negociação futura.

Agree

Esta transição é disponível ao replicante a partir dos estados *offered* e *proposed*. *Agree* indica o acordo do replicante frente a uma oferta ou proposta feita pelo outro participante. O estado *agree* estabelece uma colaboração no estado *agreed* indicando um acordo entre as partes sobre o assunto da colaboração.

Reject

A transição *reject* pode ser chamada sobre qualquer estado dentro do estado *open*. *Reject* indica a rejeição do assunto por um participante da negociação, terminando no estado *rejected*.

Timeout

A transição *timeout* é associada ao estado *open* e está ativa durante qualquer dos seguintes estados: *proposed*, *requested* ou *offered*. O *timeout* indica que o tempo da última transição, depois de decorrido, causará a execução da transição. A transição *timeout* leva ao estado *timeout*, o qual indica a falha na execução da negociação.

O modelo de negociação de [10] difere do modelo do OMG na transição inicial, chamada *CallForProposals*. Esta transição substitui todas as três transições iniciais do modelo do OMG, iniciando pelo estado *reques-*

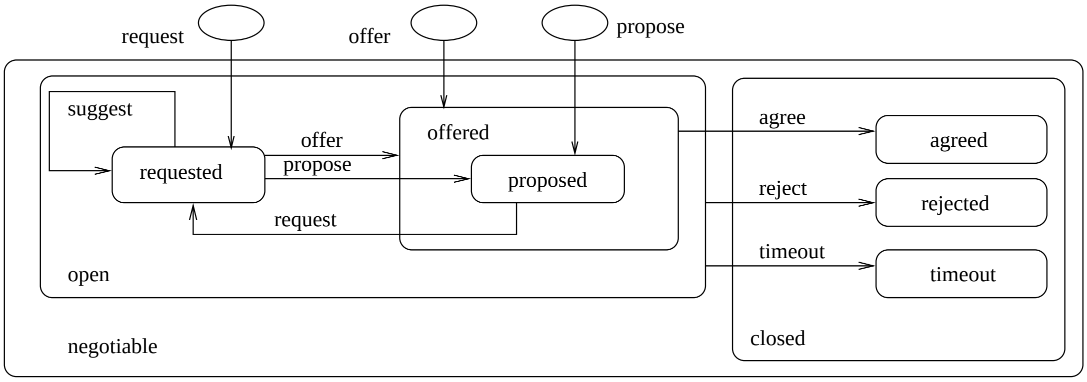

Figura 1: Modelo de Negociação Bilateral do OMG

ted. O modelo funciona da seguinte maneira: O cliente cria um agente de compra chamado de original, contendo as características do produto que deseja adquirir e o preço máximo disposto a pagar por ele. Este agente se replica de forma que várias instâncias do mesmo agente, chamadas de agentes de compra, possam negociar com vários vendedores de forma simultânea. Por sua vez, cada vendedor possui um agente de venda que vai negociar com um agente de compra, e que contém um preço mínimo pelo qual ele pode vender o seu produto. Após algumas opções serem descartadas, ou pela falta do produto ou de alguma característica, ou por não se ter chegado a um acordo sobre o preço do produto, os agentes de compra voltam com as ofertas para o agente de compra original, que as repassa para o cliente para que ele decida pela compra.

Antes de explicar como este protocolo funciona, é importante explicar aqui dois conceitos: peso e similaridade.

O que vai definir com quais agentes de venda o agente de compra irá negociar são as características requeridas do produto. Quando o usuário define as características desejadas, para cada uma, ele define também um peso que varia de 1 a 10. Quanto maior o peso, mais importante esta característica é. Depois é calculado o peso total do produto, que é a somatória dos quadrados dos pesos de cada característica. Os pesos são elevados ao quadrado para enfatizar as características mais importantes definidas pelo usuário. Já foram feitos questionamentos a respeito de elevar os pesos ao cubo ou até mesmo para valores mais altos, porém as simulações feitas em [10] mostraram que os produtos mais similares ao procurado pelo usuário foram encontrados ao elevar os pesos ao quadrado.

O outro conceito é o de similaridade. Similaridade define quão semelhante um produto é do outro baseado no peso total de suas características. O trabalho [10] utilizou similaridade apenas na escolha dos vendedores para se negociar. Quando o agente de venda busca em seu catálogo pelo produto desejado, ele verifica quais das características que o usuário quer o produto possui. Para cada característica encontrada, o mesmo peso definido pelo usuário é atribuído ao produto do vendedor, calculando também um peso total da mesma forma que o anterior. Se o peso total do produto no vendedor for maior ou igual à metade do peso total do produto desejado pelo usuário, este vendedor é escolhido pelo agente de compra original para negociar. Neste trabalho pretendemos utilizar o conceito também para a efetivação da compra.

O protocolo de negociação de [10] é dividido em 5 etapas:

- Fase da Chamada

  Nesta fase acontece o envio de uma mensagem *multicast* por parte do agente de compra para os agentes de venda. Esta mensagem é uma chamada por propostas, onde os agentes de venda podem aceitá-la ou rejeitá-la e é representada pela transição *CallForProposals*. Esta mensagem carrega as características do produto e os seus pesos. O vendedor, ao recebê-la, compara com os produtos que possui em seu catálogo para saber se possui o produto requerido e com as características desejadas. O agente de venda pode responder com *propose*, *offer* ou *rejected*. As duas primeiras são utilizadas para enviar repostas ao comprador, contendo os pesos totais de cada produto encontrado. A última para rejeitar o pedido do cliente. Aqui o comprador pode disparar também a transição *timeout* caso o agente de venda demore em responder.

- Fase da Seleção

  Nesta fase o agente de compra seleciona, dentre os agentes de venda que responderam afirmativamente na fase anterior, aqueles com os quais ele irá negociar, baseado no peso dos produtos, como explicado anteriormente. Esta seleção é formalizada pela transição *request*, a qual funciona como uma contrapoposta da chamada lançada na primeira fase.

Se o agente de venda responder com uma transição *offer* e caso o agente de compra vislumbre um ganho nesta proposta, ela será armazenada para ser apresentada ao usuário na fase de Apresentação.

- Fase da Negociação

  Nesta fase uma cópia do agente de compra original é enviada para cada *host* dos agentes de venda selecionados na fase anterior. Aqui então ocorre a negociação do preço do produto. Isto é feito através de propostas e contrapropostas entre os dois agentes. O agente de compra vai gradativamente aumentando a sua proposta, iniciando com um preço um pouco abaixo do desejado pelo cliente até o máximo disposto a pagar. Por sua vez o agente de venda começa oferecendo o preço de catálogo e vai diminuindo até o menor preço permitido. O acordo entre os dois acontece quando o preço ofertado pelo agente de venda é menor que o proposto pelo agente de compra. Se isto não acontecer a negociação é encerrada quando o agente de compra teria que propor um preço maior que o máximo que o cliente se propõe a pagar. As propostas e contrapropostas são representadas pelas transições *offer* e *propose*. Já as transições *request* e *suggest* são utilizadas para invocar por novas propostas.

- Fase da Apresentação

  Nesta fase acontece a apresentação das propostas finais, entre as negociações dos agentes de compra e venda, para o agente de compra original. E este, por sua vez, as apresenta para o cliente.

- Fase da Conclusão

  Nesta fase acontece o fechamento da negociação, onde o consumidor escolhe uma das propostas apresentadas anteriormente. Aqui pode acontecer também a abertura de uma nova sessão de negociação por parte do consumidor, através da mudança do preço do produto, ou de alguma de suas características.

3 Modelo de Negociação Proposto para Comércio Móvel

O modelo de negociação estendido para comércio móvel é apresentado na Figura 2. Um dos pontos em que ele difere do modelo anterior é na criação de uma nova mensagem chamada *CallForNegotiation*. Esta mensagem é utilizada para representar o desejo do usuário de iniciar uma nova negociação. Ela foi criada devido ao usuário estar em um dispositivo móvel, sujeito à desconexão. Desta forma, ele pode iniciar uma negociação e se desconectar sem perder a negociação corrente. Outra diferença é a criação de uma mensagem chamada *Discount*. Esta mensagem sempre é utilizada uma vez, após os participantes chegarem a um acordo, como uma contraproposta final do agente de compra para o agente de venda. Ela é utilizada caso o vendedor esteja usando algum programa de fidelidade e queira premiar aquele comprador.

O modelo funciona da seguinte maneira: Cada vez que o usuário de rede móvel deseja realizar uma compra, uma aplicação chamada *BuyApplication*, existente no dispositivo móvel, se conecta com um servidor em uma rede fixa, onde há outra aplicação chamada *CallApplication*. Esta última cria um agente de compra original para cada requisição recebida. Há pelo menos uma *CallApplication* em cada rede fixa. A Figura 3 ilustra este modelo. Setas cheias representam troca de mensagens, enquanto que setas tracejadas representam tanto troca de mensagens quanto criação de agentes. A Figura 3(a) mostra a criação de n agentes de compra originais devido a n requisições de compra. Por sua vez cada agente de compra original cria certa quantidade de cópias de agentes de compra para negociar com agentes de venda. A proporção entre agentes de venda e de compra é de um para um. A Figura 3(b) ilustra que o agente de compra original A1 criou duas cópias de agentes de compra que negociam com dois agentes de venda para atender uma requisição recebida. Um agente de compra negocia com apenas um agente de venda.

Umas das características inerentes de aparelhos móveis é a sua personalização. Por isto é interessante a criação de um perfil de usuário para ser utilizado no comércio móvel. Neste trabalho criamos dois tipos de perfis. O primeiro é o perfil do usuário propriamente dito. Cada item do perfil influencia de alguma forma na negociação e eles não são alterados com muita frequência, e uma vez definidos, eles serão válidos para qualquer transação que o usuário venha a fazer. Este perfil é caracterizado pelos seguintes itens:

- *Number of Negotiations*

  Define o número máximo de negociações realizadas em uma transação, ou seja, o número máximo de cópias de agentes de compra criados para negociar. Ao iniciar uma negociação o agente original de compra realiza uma busca por agentes de venda. Então, ele seleciona uma quantidade de agentes de venda, definida em *Number of Negotiations*, e cria esta mesma quantidade de cópias suas enviando-as uma para cada agente de venda para realizar a negociação. Se *Number of Negotiations* for definido como 0 (zero), então serão criados tantos agentes de compra quanto forem o número de agentes de venda que responderam.

- *Number of Results*

  Define o número máximo de respostas desejadas, ou seja, quantas respostas retornarão ao usuário.

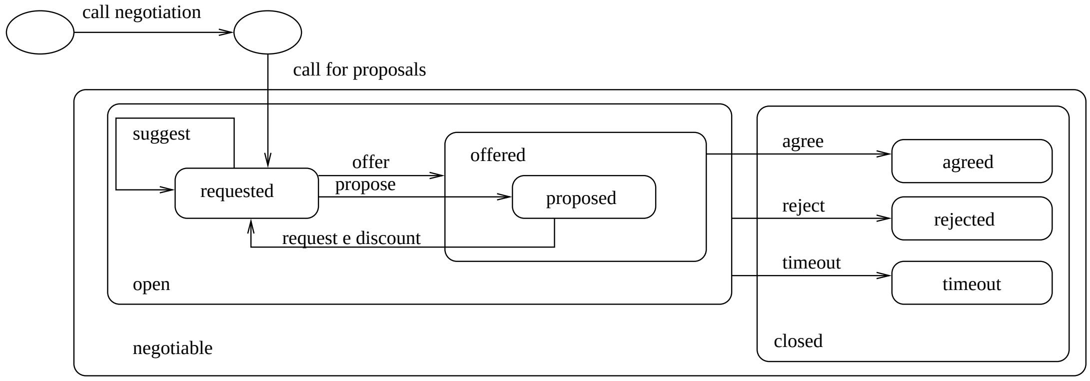

Figura 2: Modelo de Negociação Estendido para Comércio Móvel

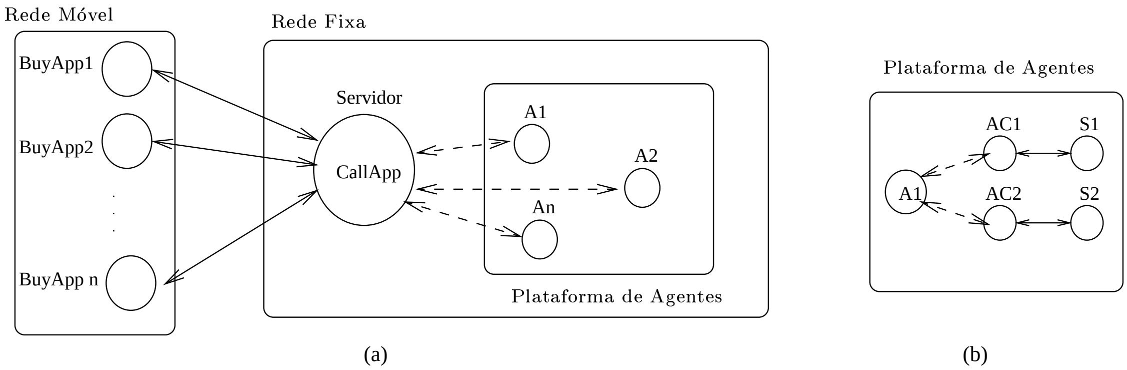

Figura 3: Comunicação do Modelo

- *Automatic Purchase*

  Ao retornar todas as respostas ao agente de compra original, define se este agente deve efetuar a compra, ou não, pelo menor preço negociado sem antes consultar o usuário. Desta forma o usuário delega ao agente a responsabilidade de realizar a compra em seu nome, criando desta forma agentes com uma maior capacidade de decisão. Depois, o usuário é avisado sobre a compra.

O outro tipo de perfil é o da compra. Neste perfil o usuário estabelece os valores dos itens que compõem o perfil a cada transação que realiza. Este perfil é caracterizado pelos seguintes itens:

- *Negotiation Weight*

  Este item define se o peso das características do produto será levado em conta na hora de escolher o produto. Se não estiver habilitado, será levado em conta apenas o seu preço. Porém, como explicado anteriormente, o peso é sempre considerado na fase de Seleção.

- *Fast Negotiation*

  Aqui o usuário define se ele quer uma negociação rápida. Se este item estiver habilitado, o agente de compra original efetuará a compra do primeiro resultado que estiver abaixo do preço desejado mais uma porcentagem (*Variation*). Este item assume que o agente irá efetuar a compra, não precisando informar ao usuário sobre os resultados da negociação. No nosso modelo este item se sobrepõe ao item *Automatic Purchase* do perfil do usuário, ou seja, não importa se *Automatic Purchase* está habilitado ou não, aqui a compra é realizada. Outra alternativa seria desacoplar estes dois itens, fazendo com que o usuário pudesse selecioná-los separadamente.

- *Fidelity*

  Este item é utilizado se o usuário deseja manter fidelidade a um vendedor. Ao retornarem todas as repostas, o agente de compra original procura pela oferta do vendedor mencionado. Se a proposta dele for menor que o preço desejado mais uma porcentagem (*Variation*), o agente realiza a compra, senão ele retorna todos os resultados para o usuário para que este se decida pela compra. Este item também se sobrepõe ao item *Automatic Purchase*, existindo a alternativa de desacoplá-los também, assim como no item anterior.

- *Connected*

  Este item é habilitado se o usuário deseja manter a conexão com o servidor enquanto a negociação é realizada. Se isto acontecer o agente de compra original informa ao usuário o resultado. Se não estiver habilitado, é gravado em um arquivo informação de que há uma negociação pendente, junto com o endereço do servidor e o identificador do agente de compra original que realizou a negociação. Após isso, o usuário é desconectado. Na próxima vez que o usuário conectar, ele verifica se há uma negociação pendente. Se houver, a conexão é estabelecida com o agente de compra original utilizando as informações contidas no arquivo e ocorre a solicitação dos resultados.

- *Variation*

  Define a porcentagem, em cima do preço desejado, que o agente de compra pode oferecer a mais nas negociações realizadas levando em conta os itens *Fast Negotiation* e *Fidelity* do perfil da compra. Este item foi criado com o intuito de alcançar uma negociação mais semelhante à que acontece na vida real. O agente de compra tenta conseguir o preço desejado pelo usuário, porém se não consegue, ele aceita uma margem (*Variation*) de preço um pouco maior, ao relizar uma negociação com *Fast Negotiation* ou *Fidelity*.

O protocolo estendido para comércio móvel está dividido em 7 fases. Explicaremos a seguir o que foi estendido em cada fase para comércio móvel. O que não for explicado, permanece o mesmo que em [10], sendo que este não possui as fases de conexão e análise.

- Fase da Conexão

  Há no aparelho portátil uma aplicação que irá iniciar a negociação, chamada de *BuyApplication*. O usuário ao requisitar uma nova negociação, *BuyApplication* comunica com o servidor que hospeda uma aplicação chamada de *CallApplication* que espera por requisições na rede fixa. A função desta aplicação é servir de interface entre o dispositivo móvel e o ambiente de agentes móveis (Grasshopper, neste trabalho). Quando acontece uma requisição, esta aplicação cria um agente de compra original (*BuyAgent*) passando como argumentos os atributos recebidos pela *BuyApplication*. Tais atributos são:

  - Características do produto e seus respectivos pesos.

    Como explicado na Seção 2, estes pesos são utilizados para o cálculo de similaridade.
  - O preço máximo e o preço que o usuário gostaria de pagar pelo produto.

    Estes dois valores são utilizados para se fazer a negociação de preços com o agente vendedor.
  - O perfil do usuário
  - O perfil da compra

  Após o agente de compra original ser criado, *CallApplication* enviará uma mensagem para *BuyApplication* contendo a identificação deste agente associado à sua negociação, para que o mesmo possa, no caso de desconexão, ser encontrado.

- Fase da Chamada

  O mesmo que no protocolo de negociação para comércio eletrônico.

- Fase da Seleção

  Nesta fase além do peso dos produtos, o número máximo de negociações definido no perfil do usuário também é levado em conta na hora de selecionar com quais vendedores o agente de compra irá negociar.

- Fase da Negociação

  Nesta fase, se chegarem ao acordo, o agente de compra ainda requisita um desconto ao agente de venda em cima do preço conseguido através do método *Discount*. O agente de venda pode conceder ou não o desconto, dependendo se ele utiliza algum serviço de fidelidade e se quem solicitou o desconto é ou não seu cliente. Aqui também ocorre o retorno dos resultados para o agente de compra original.

- Fase da Análise

  É nesta fase que as decisões são tomadas levando em consideração a maioria das informações contidas no perfil do usuário e da compra. As possibilidades são as seguintes:

  - *Fast Negotiation*. Aqui não é preciso que todas as respostas retornem. Cada resposta recebida é imediatamente analisada pelo agente de compra original. Se o resultado obtido for menor que o preço desejado mais uma porcentagem (definida no perfil do usuário pelo campo *Variation*), o agente realiza a compra. Senão ele compra do mais barato. Este tipo de negociação é extremamente útil no comércio móvel já que aparelhos portáteis possuem pouca autonomia devido à sua dependência de baterias;
  - *Fidelity*. Com esta opção habilitada, o agente de compra original espera todas as repostas retornarem e então procura se há entre elas alguma que venha do vendedor com o qual o usuário deseja manter fidelidade. Se houver

e a oferta do vendedor for menor que o preço desejado mais uma porcentagem (*Variation*), a compra é realizada. Senão ele retorna todos os resultados para o usuário para que ele decida pela compra. Vale salientar que estas duas últimas opções consideram que o usuário deseja realizar a compra, mesmo que a opção *Automatic Purchase* não esteja habilitada;

  - *Negotiation Weight*. Ao retornar todos os resultados, o agente de compra original verifica se esta opção está habilitada. Se estiver, ele ordena os resultados por um peso relativo. Este peso relativo é conseguido através da divisão do peso do produto pelo seu preço, ambos retornados de cada vendedor. Isto é feito se o cliente quiser levar em consideração as características na hora de realizar a compra, uma vez que, para que um vendedor seja selecionado, basta que o peso do produto no seu catálogo seja maior ou igual à metade do peso que o usuário definiu. Isso evita que o cliente compre o produto mais barato e no entanto com características mínimas exigidas por ele. Por exemplo, podem retornar dois resultados, x e y, sendo x levemente mais caro do que y, no entanto exatamente como o usuário deseja. Neste caso o usuário poderia se decidir por x utilizando a opção *Negotiation Weight*.
  - *Automatic Purchase*. Se esta opção estiver habilitada, o agente de compra original espera todas as repostas retornarem e após isso ele envia uma mensagem para a sua cópia que conseguiu o preço mais barato e ordena a compra do produto. Se não estiver habilitada, ele ordena as ofertas em ordem crescente de preço e as envia para a aplicação que se encontra no dispositivo móvel, para que esta as apresente ao usuário, respeitando o limite de resultados definidos pelo mesmo;

- Fase da Apresentação

  Esta fase começa quando o agente de compra original possui os resultados e já tomou todas as decisões sobre estes baseado no perfil do usuário e da compra. Se a opção *Connected* estiver habilitada, ele envia os resultados para *BuyApplication*, através de *CallApplication,* para serem apresentados ao usuário. Senão, ele aguarda até que o usuário estabeleça conexão e solicite os resultados. O mesmo ocorre quando o agente de compra original decide pela compra sem consultar o usuário. Neste caso, uma mensagem, dizendo que a compra foi realizada e onde foi realizada, é enviada para o usuário.

- Fase da Conclusão

  Nesta fase ocorre o fechamento da negociação, onde o consumidor escolhe uma das propostas apresentadas anteriormente ou apenas toma conhecimento de que a compra foi realizada e onde. Aqui pode acontecer também a abertura de uma nova sessão de negociação por parte do consumidor, através da mudança do preço do produto, ou de alguma de suas características.

A seguir são apresentados diagramas de sequência com o fluxo de mensagens entre a aplicação no dispositivo móvel (*BuyApplication*), a aplicação do servidor (*CallApplication*), o agente de compra original (*BuyAgent*) e os agentes de compra (*BuyAgentC0...n*) e venda (*SellAgentS0...n*). As setas cheias representam as transições do modelo. As setas tracejadas representam chamadas da plataforma de agentes móveis. As setas pontilhadas representam transições que não alteram o estado do modelo.

A Figura 4 representa a sequência de mensagens trocadas durante uma negociação em que o usuário permanece conectado esperando a reposta do agente de compra original, e que não é utilizada nenhuma forma de compra automática (*Automatic Purchase*, *Fast Negotiation* e *Fidelity*), definidas nos perfis do usuário e da compra. As mensagens *CallForNegotiation* e *CreateAgent*, utilizada por *CallApplication* para criar o agente de compra original, ocorrem na fase de Conexão. Na fase de Chamada é disparada a mensagem *CallForProposals* do agente de compra original para os agentes de venda. O retorno desta mensagem é utilizado pelo agente de compra original para selecionar para quais agentes de venda ele irá enviar cópias de agentes de compra (fase de Seleção). Na fase de Negociação há a cópia dos agentes de compra para os *hosts* dos agentes de venda, através de *CopyAgent*, a negociação de preços (*BidPrice* e *Discount*) e o retorno dos resultados para o agente de compra original através de *ReturnResult*. Vale ressaltar que no diagrama de sequência, *BidPrice* substitui as transições *offer*, *propose* e *request* do diagrama de estados mostrado anteriormente. O que vai definir em que estado a negociação se encontra é o valor de retorno do método *BidPrice*. Na fase de Análise não há troca de mensagens. Todas as decisões são tomadas pelo agente de compra original. Após tomar as decisões, o agente de compra original envia os resultados para o *BuyApplication* através de *CallApplication* utilizando a mensagem *ReturnResultU* (fase de Apresentação). Por fim, na fase de Conclusão, o usuário decide, ou não, pela compra enviando a mensagem *Response* para o agente de compra original, e este por sua vez, informa aos agentes de compra, através de *Agree* e *Reject*, quem irá comprar o produto, se o usuário houver decidido pela compra.

A Figura 5 é utilizada para representar a negociação

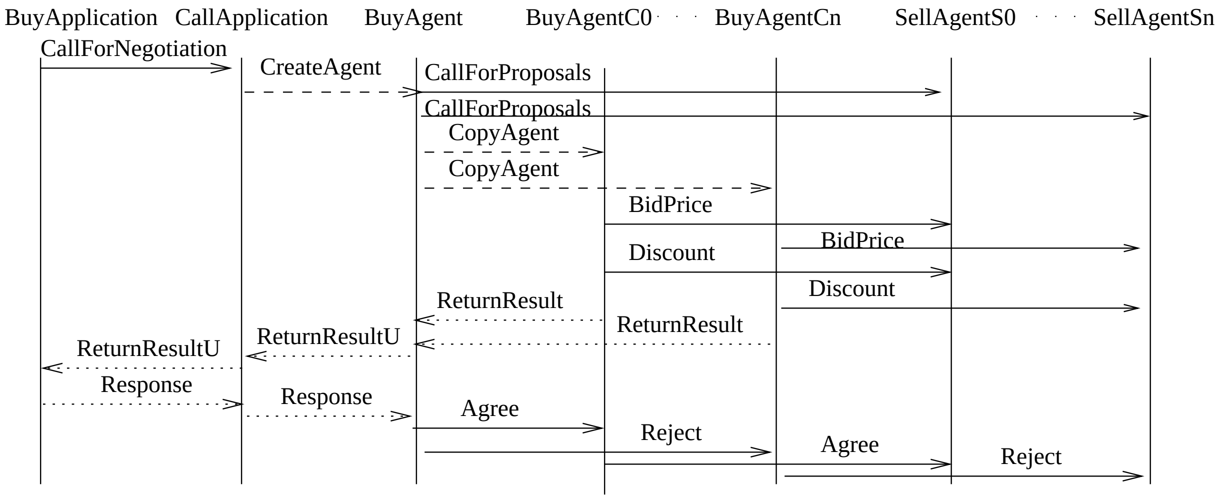

Figura 4: Diagrama de Sequência em Negociação Conectada

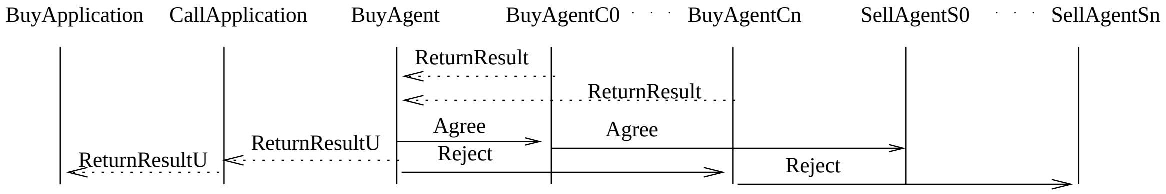

Figura 5: Diagrama de Sequência em Negociação com Compra Automática

quando é utilizada alguma forma de compra automática. Tudo acontece da mesma maneira que no diagrama anterior até o momento do retorno dos resultados pelo agente de compra para o *BuyAgent*, denotado pela mensagem *ReturnResult*. A partir daí *BuyAgent* decide o que fazer com base nos perfis do usuário e da compra. No caso da Figura 5, ele aceita a proposta de um dos agentes de venda e recusa a dos outros, realizando a compra e informando ao usuário através de *ReturnResultU*.

Por último, a Figura 6 representa a sequência de mensagens que ocorre quando acontece uma negociação em que o usuário se desconecta da rede fixa e se reconecta mais tarde, solicitando os resultados. A sequência de troca de mensagens é semelhante às anteriores, até o retorno dos resultados pelo agentes de compra para o *BuyAgent*. A mensagem não prossegue pois o usuário está desconectado. Ao se reconectar, *BuyApplication* solicita os resultados a *BuyAgent* através de *CallApplication* utilizando a mensagem *RequestResult*. Os valores de retorno desta mensagem são os resultados da negociação.

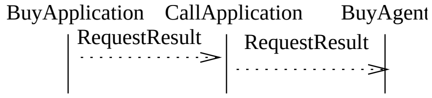

Figura 6: Diagrama de Sequência em Negociação Desconectada

4 Aspectos de Implementação

Foi desenvolvido um protótipo para validar o modelo de negociação descrito anteriormente. Os agentes foram implementados usando o Java Developmemt Kit 1.2.2 (JDK 1.2.2) e a aplicação no dispositivo móvel foi implementada usando o Java 2 Micro Edition Wireless Toolkit versão 1.0.4 (J2ME). O sistema de agentes usado para fornecer a infra-estrutura de comunicação foi o Grasshopper versão 2.1 [3]. Sua escolha foi devido por este fornecer interoperabilidade entre diferentes sistemas de agentes, através do seu suporte a interface MASIF (*Mobile Agent System Interoperability Facility*) [6]. Os catálogos dos vendedores foram implementados em XML.

O Grasshopper é composto por agências (*agency*), regiões (*regions*) e lugares (*places*). Uma agência é o ambiente de execução para agentes móveis e estacionários. Um lugar é um agrupamento lógico de funcionalidades dentro de uma agência. Em uma agência podem existir vários lugares. As agências, bem como seus lugares, podem ser associados a uma região.

Os agentes de venda podem estar ativos em qualquer agência, contudo eles precisam estar na mesma região do agente de compra original, uma vez que ele realiza a busca por agentes de venda através do nome em uma região específica.

A comunicação entre agentes é realizada através do serviço de comunicação disponível no sistema de agentes, utilizando a criação de *proxies* assíncronos, para que o agente não fique bloqueado esperando uma eventual resposta. As chamadas de métodos são realizadas apenas nos métodos da interface de cada agente, as quais são apresentadas na Figura 7. Vale salientar que na interface *IBuyAgent* alguns métodos são chamados apenas sobre o agente de compra original, enquanto outros sobre a cópia do agente de compra.

```
IBuyApplication
public void ReturnResultU (String[ ] vetProfile)

ICallApplication
public Identifier CallForNegotiation (String vetProfile)
public void ReturnResultU (String[ ] vetReturn, int n)
public String[ ] RequestResult( )
public void Response (String[ ] vetResponse)

IBuyAgent
public void ReturnResult (long price, Identifier agentId,
                                    GrasshopperAddress address)
public String[ ] RequestResult( )
public void Response (String[ ] vetResponse)
public void Agree ( )
public void Reject( )

ISellAgent
public String[ ] CallForProposals (String[ ] vetArg)
public String BidPrice (Identifier agentId, long priceBidden)
public long Discount (String userId)
public GrasshopperAddress Agree (Identifier agentId)
public String Reject (Identifier agentId)
```

Figura 7: Métodos das Interfaces

O método *ReturnResult* de *IBuyApplication* e *ICallApplication* é utilizado para retornar os resultados da negociação para as aplicações *BuyApplication* e *CallApplication*, respectivamente. *RequestResult* de *ICallApplication* e *IBuyAgent* é utilizado quando há uma negociação em que o usuário se desconecta. Através deste método ele requisita os resultados ao se conectar. Os métodos *ReturnResult*, *RequestResult* e *Response* de *IBuyAgent* são chamados apenas sobre o agente de compra original, enquanto que os métodos *Agree* e *Reject* apenas nos agentes de compra. *BidPrice* e *Discount* são chamados pelo agente de compra sobre o agente de venda e são utilizados pelos dois como mecanismo de proposta e contraproposta. *AcceptProposal* e *RejectProposal* de *ISellAgent* são chamados pelos agentes de compra sobre os respectivos agentes de venda para informar se a proposta foi aceita ou recusada.

A cada chamada de uma aplicação *BuyApplication*, *CallApplication* cria uma thread que é responsável por cuidar da negociação desta requisição, uma vez que *CallApplication* pode receber chamadas de vários usuários.

Uma vez que *BuyApplication* é executada no dispositivo móvel, ela deve ser uma aplicação simples, devido aos poucos recursos do dispositivo. Por outro lado *CallApplication* e *BuyAgent* comportam a maior parte da complexidade do modelo, já que estes executam na rede fixa, onde há mais recursos disponíveis.

A cópia do agente de compra para o *host* do vendedor é realizada através da chamada ao método *copy()* da classe *Agent* do Grasshopper. Este método requer, como parâmetro, o endereço para onde a cópia deve ir.

A aplicação para o dispositivo móvel, implementada com o J2ME, foi testada utilizando um emulador de celular presente no próprio kit de desenvolvimento. O emulador escolhido foi o do telefone celular Motorola i85s. A seguir são apresentadas algumas telas desenvolvidas.

A Figura 8 apresenta a tela principal da aplicação, onde o usuário pode escolher entre gerenciar o perfil do usuário, o perfil da compra, iniciar uma nova compra ou visualizar os resultados de uma negociação realizada anteriormente.

Figura 8: Tela Principal

A Figura 9 mostra algumas telas quando o usuário escolhe a opção *User Profile* da figura anterior. Na tela da Figura 9(a) ele pode escolher entre visualizar ou alterar o seu perfil. A tela da Figura 9(b) apresenta o perfil. Na tela da Figura 9(c), o usuário escolhe qual perfil deseja alterar, e por fim a tela da Figura 9(d) mostra a alteração do perfil *Automatic Purchase*.

A Figura 10 é similar a do perfil do usuário, contudo aqui o usuário gerencia o perfil da compra.

A Figura 11(a) apresenta a tela de uma nova compra, onde o usuário escolhe o produto que deseja comprar, neste caso um bilhete de passagem aérea. Na Figura 11(b) o usuário escolhe uma característica do produto. A Figura 11(c) mostra o usuário definindo a característica data do vôo. A Figura 11(d) apresenta os resultados retornados de uma negociação, enquanto que a Figura 11(e) apresenta o resultado detalhado da opção 1.

(a) (b) (c) (d)

Figura 9: Telas de Gerenciamento do Perfil do Usuário

(a) (b) (c) (d)

Figura 10: Telas de Gerenciamento do Perfil da Compra

5 Conclusão

Este trabalho apresenta um modelo de negociação automatizada, com agentes móveis, para comércio móvel. Existem alguns trabalhos sobre negociação automatizada em comércio eletrônico, como [7] e [1]. O primeiro propõe um modelo de negociação orientada a serviço. O segundo descreve um sistema multi-agente para comércio eletrônico e avalia diferentes estratégias de negociação de agentes. Outros trabalhos mostram o uso de agentes móveis em comércio móvel. [8] propõe o uso de agentes móveis para lidar com a questão de identificação do usuário e do dispositivo pessoal e realização de transações utilizando um pequeno números de mensagens. [4] propõe uma aplicação dedicada a usuários móveis usada para gerenciar seus agentes móveis para realizar tarefas em nome do usuário.

O modelo deste trabalho utiliza agentes de compra e de venda para realizar a negociação de preços. Cópias do agente de compra original se movem para os *hosts* dos vendedores para negociar com os respectivos agentes de venda, de forma autônoma e independente em benefício do usuário. Uma contribuição do trabalho foi a criação e a utilização de um perfil do usuário de dispositivo móvel na negociação automatizada. Este perfil é utilizado na fase de Análise para se tomar decisões com relação à compra ou não de produtos. Um exemplo de item deste perfil é *Automatic Purchase*, que define se o agente de compra original deve realizar a compra ao encontrar o produto requerido com o preço desejado sem consultar o usuário antes. Foi utilizado também um perfil de compra, onde o usuário define características referentes a apenas aquela transação. Este perfil também é utilizado na fase de Análise. Como exemplo podemos citar o item *Connected* que decide se *BuyApplication* permanecerá, ou não, conectada a *CallApplication* esperando o resultado da negociação. Vale lembrar que estes dois perfis são flexíveis, de maneira que é possível serem inseridos novos itens para serem avaliados, durante a tomada de decisões da negociação. Outra contribuição é o fato de se levar em conta a medida de similaridade na hora de se escolher o produto a comprar através de uma relação entre a similaridade do produto encontrado e o preço oferecido.

Vale lembrar que a aplicação residente no celular é simples devido à pouca quantidade de recursos disponíveis no dispositivo móvel. Já as aplicações da rede fixa possuem maior complexidade, o que não é um problema, pois a rede fixa possui maior quantidade de recursos. Outra vantagem deste protocolo é que ele não sobre-

(a) (b) (c) (d) (e)

Figura 11: Telas de uma Compra Nova e Visualização dos Resultados

carrega os recursos limitados do canal de comunicação devido a pouca quantidade de dados trafegados da rede móvel para a rede fixa e vice versa, e também da não necessidade de conexão do usuário durante a negociação.

Uma vantagem de utilizar agentes móveis para a negociação é que eles realizam várias negociações em paralelo, além do que o cliente não fica bloqueado esperando o retorno dos resultados, podendo realizar outras tarefas. Outra vantagem é a minimização da troca de mensagens na rede fixa entre os agentes de compra e venda, uma vez que os primeiros se movem para os *hosts* dos agentes de venda.

O Grasshopper foi escolhido como infra-estrutura para os agentes devido ao seu suporte a interface MASIF, que fornece interoperabilidade entre diferentes sistemas de agentes. Foram apresentadas algumas telas da aplicação que é executada no dispositivo móvel. As telas foram desenvolvidas levando em conta os poucos recursos dos dispositivos móveis.

Como trabalho futuro, pretendemos elaborar um mecanismo de interação entre agentes, de tal forma que as cópias dos agentes de compra possam se comunicar entre si para que elas possam tomar decisões já na negociação, baseadas nas informações das outras cópias e para que retornem o menor número possível de informações para o agente de compra original. Outro trabalho futuro será a criação de um mecanismo de pagamento no modelo utilizando-se também os perfis criados neste trabalho. Um exemplo de perfil é a forma de pagamento: à vista ou a prazo.

Referências

[1] Henrique Lopes Cardoso, Max Schaefer e Eugenio Oliveira. A multi-agent system for electronic commerce including adaptative strategic behaviours. In *Encontro Português de Inteligência Artificial - EPIA 99*, Évora, Portugal, Setembro 1999.

[2] Anthony Chavez e Pattie Maes. Kasbah: An agent marketplace for buying and selling goods. In *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, pages 75–90, Londres, Inglaterra, Abril 1996.

[3] IKV. Grasshopper - The agent plataform. http://www.grasshopper.de/.

[4] Alain Macaire, David Carlier e Pierre Paradinas. A personal agent manager for mobile users. In *European Multimedia, Embedded Systems, Technologies for Business Processes and Electronic Commerce - EMMSEC 99*, Estocolmo, Suécia, Junho 1999.

[5] Stephen McConnell. *Negotiation Facility - Final Revised Submission.* OMG, Março 1999. http://www.osm.net/upload/99-03-06.pdf.

[6] OMG. Masif. http://cgi.omg.org/cgi-bin/doc?orbos/98-03-09.pdf.

[7] Carlos Sierra, Peyman Faratin e Nick R. Jennings. A service-oriented negotiation model between autonomous agents. In *Proceedings of the 8th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW-97)*, pages 17–35, Ronneby, Suécia, Março 1997.

[8] Jian Tang e Jari Veijalainen. Using agents to improve security and convenience in mobile e-commerce. In *Proceedings of the 34th Hawaii International Conference on System Sciences - HICSS-34*, Hawaii, Estados Unidos, Janeiro 2001.

[9] Aphrodite Tsalgatidou, Jari Veijalainen e Evaggelia Pitoura. Challenges in mobile electronic commerce. In *Proceedings of Iec 2000. Third Internacional Conference on Innovation through E-Commerce*, pages 169–174, Manchester, Inglaterra, Novembro 2000.

[10] Jo Ueyama e Edmundo R. M. Madeira. An automated negotiation model for electronic commerce. In *Proceedings of tht Fifth IEEE Internationl Symposium on Autonomous Decentralized Systems - ISADS'01*, pages 29–36, Dallas, Estados Unidos, Março 2001.

[11] W3C. Xml. http://www.w3c.org/XML/.

[12] Shunsuke Yurai, Kazuhiro Kiwata, Atsushi Nakano, Tomotaka Uchihashi e Atsushi Kanai. Service matching and collaboration for electronic commerce. In *Proceedings of the Fifth IEEE Internationl Symposium on Autonomous Decentralized Systems - ISADS'01*, pages 169–174, Dallas, Estados Unidos, Março 2001.