

UNIVERSIDADE FEDERAL DE SANTA CATARINA

USO DE AGENTES INTELIGENTES NA BUSCA DE INFORMAÇÃO DISTRIBUÍDA

JULIA DE FARIA STAMM

Florianópolis – SC

2012/2

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

USO DE AGENTES INTELIGENTES NA BUSCA DE INFORMAÇÃO DISTRIBUÍDA

JULIA DE FARIA STAMM

Trabalho de conclusão de curso
apresentado como parte dos
requisitos para obtenção do grau
de Bacharel em Sistemas de
Informação.

Florianópolis – SC

2012/2

JULIA DE FARIA STAMM

USO DE AGENTES INTELIGENTES NA BUSCA DE INFORMAÇÃO DISTRIBUÍDA

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Orientadora: Luciana de O. Rech

Banca Examinadora

Lau Cheuk Lung

Tatiana Pereira Filgueiras

"Um homem precisa viajar. Por sua conta, não por meio de histórias, imagens, livros ou TV. Precisa viajar por si, com seus olhos e pés, para entender o que é seu. Para um dia plantar as suas árvores e dar-lhes valor. Conhecer o frio para desfrutar o calor. E o oposto. Sentir a distância e o desabrigo para estar bem sob o próprio teto. Um homem precisa viajar para lugares que não conhece para quebrar essa arrogância que nos faz ver o mundo como o imaginamos, e não simplesmente como é ou pode ser; que nos faz professores e doutores do que não vimos, quando deveríamos ser alunos, e simplesmente ir ver."

(Amir Klink)

AGRADECIMENTO

Agradeço aos meus pais Ana e Hugo por serem exemplos de vida para mim. Eles são fruto da minha inspiração e dedicação.

Ao meu irmão Bruno, por seu espírito jovem e empreendedor.

A minha orientadora Luciana, pela atenção e dedicação durante este trabalho.

SUMÁRIO

1. Introdução.....	13
1.1. Motivação	13
1.2. Objetivo Geral	14
1.3. Objetivos Específicos	14
1.4. Organização do Texto	15
2. Agentes Inteligentes	16
2.1. Introdução.....	16
2.2. Definição.....	16
2.3. Tipologia dos agentes.....	17
2.4. Sistemas Multiagentes.....	18
2.5. Desenvolvimento	20
2.5.1. Padronização FIPA	20
2.5.2. Plataforma de Agentes – JADE.....	20
3. Trabalhos Relacionados	22
3.1. Introdução	22
3.2. ShopBot [DOORENBOS; ETZIONI; WELD, 1997]	22
3.3. ShopBot da Universidade de Hanyang [CHOI; HAM; KIM; LEE; YANG, 2000]	25
3.4. Sistema de Compras Inteligentes [MENG; ZENG, 2005]	28

3.5. Assistente Pessoal de Viagens [ALHASHEL; BALACHANDRAN; MOHAMMEDIAN, 2009]	30
3.6. Considerações Finais sobre as Abordagens Envolvendo SMA na Busca de Informações em Sites de Compra.....	33
3.7. Conclusão.....	34
4. Sistema BuscaCompra	35
4.1. Introdução.....	35
4.2. Apresentação do Sistema.....	35
4.3. Requisitos	36
4.3.1. Requisitos Não Funcionais.....	36
4.3.2. Requisitos Funcionais	37
4.4. Metodologia	38
4.5. Estrutura dos Agentes	41
4.5.1. Agente Interface.....	43
4.5.2. Agente Sistema.....	44
4.5.3. Agente Loja	45
4.5.4. Agente Compra Coletiva	46
4.6. Arquitetura	47
4.7. Desenvolvimento	48
4.8. Conclusão.....	51
5. Avaliação do Sistema	52
5.1. Introdução.....	52
5.2. Testes de desempenho	52

5.2.1. Execução e Resultados dos Testes	53
5.3. Conclusão.....	56
6. Conclusões e Trabalhos Futuros.....	57
6.1. Revisão das Motivações e Objetivos	57
6.2. Principais contribuições	57
6.3. Trabalhos Futuros.....	58
Referências Bibliográficas	59

LISTA DE FIGURAS

Figura 3.1: O gerador de ontologias (CHOI; HAM; KIM; LEE; YANG, 2000).....	27
Figura 3.2: Estrutura das ontologias (CHOI; HAM; KIM; LEE; YANG, 2000)	27
Figura 3.3: A estrutura do sistema multiagente (MENG; ZENG, 2005)	29
Figura 3.4: Negociação multiservidor com um moderador (ALHASHEL; ALACHANDRAN; MOHAMMEDIAN, 2009).....	31
Figura 3.5: Componentes do sistema Assistente Pessoal de Viagens (ALHASHEL; BALACHANDRAN; MOHAMMEDIAN, 2009).....	32
Figura 4.1: Diagrama de casos de uso (CLAIRE; BAHRI; NIKRAZ, 2006).....	39
Figura 4.2: Diagrama de agentes (CLAIRE; BAHRI; NIKRAZ, 2006).....	40
Figura 4.3: Estrutura do sistema multiagentes BuscaCompra.....	42
Figura 4.4: Diagrama de classes com foco no Agente Interface.....	43
Figura 4.5: Diagrama de classes com foco no Agente Sistema.....	44
Figura 4.6: Diagrama de classes com foco no Agente Loja.....	46
Figura 4.7: Arquitetura do sistema BuscaCompra	48
Figura 4.8: Diagrama de classes das camadas de aplicação e negócio.....	49
Figura 4.9: Diagrama de sequência das camadas de aplicação e negócio.....	50
Figura 4.10: Diagrama de classes da camada de dados.....	51
Figura 5.1: Diagrama de casos de uso para teste de desempenho	53
Figura 5.2: Teste com a palavra-chave “camera”.....	55
Figura 5.3: Teste com a palavra-chave “bicicleta aro 16 full suspension”.....	55

LISTA DE TABELAS

Tabela 4.1: Tabela de Responsabilidades (CLAIRE; BAHRI; NIKRAZ, 2006)41

Tabela 5.1: Resultado da execução dos testes54

LISTA DE REDUÇÕES

ACC Agent Communication Channel

ACL Agent Communication Language

AMS Agent Management System

DF Directoy Facilitator

FIPA Foundation for Intelligent Physical Agents

IA Inteligência Artificial

JADE Java Agent Development Framework

PDU Unidades de Descrição do Produto

SMA Sistemas Multiagentes

RESUMO

A popularização da internet impulsionou a criação de um novo ramo na economia: o comércio eletrônico. Este comércio forneceu inúmeras conveniências ao consumidor, facilitando o acesso a milhares de produtos e uma variedade de lojas ao seu dispor. O lado negativo desta facilidade é o excesso de informações resultantes deste crescimento. Em face disso, muitas ferramentas e pesquisas vem sendo desenvolvidas para comprimir essas informações de acordo com o interesse do cliente. BuscaCompra é um sistema criado para consumidores poderem pesquisar produtos no comércio eletrônico, levando em consideração o atual cenário da economia: a existência de lojas virtuais e sites de compras coletivas. O sistema foi desenvolvido através do paradigma de software orientado a agentes, onde uma sociedade de agentes colaborativos é capaz de extrair informações de lojas virtuais e sites de compra coletiva em tempo real, com o objetivo de substituir o armazenamento de dados em dispositivos físicos.

Palavras-chave: Agentes inteligentes. Sistema multiagentes. Pesquisa de produtos.

1. Introdução

1.1. Motivação

O crescimento do comércio eletrônico forneceu inúmeras conveniências ao consumidor, dando acesso às informações de milhares de produtos, além de uma infinita variedade de lojas ao seu dispor. Como resultado desta combinação, a grande quantidade de informações torna cada vez mais difícil a decisão de fazer compras, diminuindo a habilidade do consumidor em coletar e analisar as informações relevantes (FENG, 2011).

Como consequência deste excesso de informações, websites foram criados para reunir a informação de diversas lojas e permitir maior visibilidade ao usuário em menos tempo e com menos esforço.

O Buscapé Company é uma plataforma digital de comércio eletrônico conhecida pela ferramenta de comparação de preços. Em 1999, a companhia criou o seu primeiro website chamado Buscapé, provendo serviços de comparação e pesquisas de produtos, tornando-se referência internacional de empreendedorismo digital brasileiro (ZUINI, 2011).

Em 2010, surge um novo conceito de vendas virtuais no Brasil, os sites de compras coletivas. Eles foram criados com o objetivo de oferecer descontos em produtos ou serviços exclusivos para cada cidade. Estes descontos são oferecidos apenas durante um determinado período ou quando o número máximo de vendas é atingido (FAUST, 2010).

Podemos observar dois modelos de comércio eletrônico distintos. O site Buscapé oferece ferramenta para comparação de produtos em diversas lojas, enquanto os sites de compra coletiva oferecem descontos em diversos serviços e produtos. Ambos os modelos possuem o mesmo objetivo: oferecer ao consumidor preços menores.

Diante desses dois modelos de comércio eletrônico, surge a grande motivação deste trabalho: criar uma ferramenta que possua os dois modelos de

comércio eletrônico. Um sistema que permita pesquisar produtos tanto em lojas virtuais, como em sites de compra coletiva.

O maior desafio de juntar estas informações é a imprevisibilidade das ofertas e produtos disponíveis em sites de compra coletiva. O motivo desta imprevisibilidade é porque muitas vezes um produto ou um serviço não estão mais disponíveis depois que uma determinada quantidade de compras é atingida ou depois de um determinado período.

Para aumentar a confiabilidade das informações presentes na ferramenta proposta, é necessário criar uma infraestrutura que realize a busca em sites de compra coletiva no momento em que o usuário realiza a pesquisa no sistema, pois as chances do desconto ter acabado são menores. Desta forma, ao invés de utilizar o armazenamento de dados físicos, é necessário criar uma infraestrutura que acesse os dados de cada site em tempo real.

Este trabalho propõe o desenvolvimento de um sistema multiagentes (SMA) para enfrentar este desafio. A utilização de uma sociedade de agentes colaborativos, capazes de extrair informações de lojas virtuais e sites de compra coletiva em tempo real, substituindo o armazenamento em dispositivos físicos.

1.2. Objetivo Geral

O presente trabalho tem como objetivo desenvolver um sistema multiagente utilizando agentes autônomos para a busca de informações em um ambiente distribuído.

1.3. Objetivos Específicos

1. Estudo da literatura na área de agentes inteligentes e SMA;
2. Identificar e compreender a plataforma mais conceituada para o desenvolvimento de SMA;

3. Analisar trabalhos já desenvolvidos na área de busca de informações distribuídas utilizando agentes inteligentes para a coleta de informações em sites de compra;
4. Implementação do SMA;
5. Análise de desempenho do SMA.

1.4. Organização do Texto

Este trabalho está organizado em seis capítulos, sendo o presente capítulo responsável por apresentar as informações gerais sobre o trabalho, a nível introdutório, mostrando a motivação e os objetivos esperados.

No segundo capítulo, são apresentados conceitos e definições importantes sobre agentes inteligentes, os quais serão abordados durante o trabalho. O texto apresenta: definições e tipologias do agente, definição de SMA, padronização e plataforma para o desenvolvimento de agentes.

O estudo das literaturas que envolvem SMA para extração de informações em sites de compra está presente no capítulo 3.

A modelagem e o desenvolvimento do SMA proposto por este trabalho é apresentada no capítulo 4. O objetivo é apresentar o sistema como um todo, através de: descrições, requisitos, metodologias utilizadas para sua modelagem, definição da arquitetura utilizada e o seu desenvolvimento.

O capítulo 5 apresenta a avaliação do comportamento do sistema através de testes de desempenho.

Por fim, o capítulo 6 apresenta as conclusões e os trabalhos futuros.

2. Agentes Inteligentes

2.1. Introdução

Etimologicamente, o termo Humano significa “homem sábio”. Esta definição evidencia a importância da capacidade mental para nós seres humanos. Durante muito tempo, procurou-se explicar como a matéria pode: compreender, perceber, prever. A área de inteligência artificial, ou IA, tenta não apenas compreender, mas construir máquinas inteligentes (RUSSEL; NORVIG, 2004).

A ciência de IA pode ser aplicada a uma variedade de subcampos: jogos de xadrez, robôs, demonstração de teoremas matemáticos, diagnóstico de doenças. Esta variedade só é possível porque a ciência busca sistematizar e automatizar as tarefas intelectuais, tornando-se relevante para qualquer esfera da atividade intelectual humana (RUSSEL; NORVIG, 2004).

Uma das tecnologias utilizadas para aplicação desta ciência é chamada de agentes inteligentes. Estes são entidades computacionais com a capacidade de tomar decisões e agir por conta própria. Este capítulo aborda as características dos agentes para distingui-los de programas, apresenta como são feitos sistemas constituídos por agentes e tecnologias existentes para auxiliar o desenvolvimento destes sistemas.

2.2. Definição

Um agente é considerado uma entidade computacional com comportamento autônomo, de modo que possa tomar suas próprias decisões (ALVARES; SICHMAN, 1997). Estas decisões levam em consideração o ambiente em que eles estão, fazendo necessário mecanismos de interação entre agente e ambiente. O agente é capaz de perceber o ambiente através de sensores e agir sobre o mesmo por intermédio de atuadores (RUSSEL; NORVIG, 2004). Além disso, agentes possuem características que os diferem de programas, listados abaixo (ETZIONI; WELD, 1995):

- **Autonomia:** Capacidade de decidir a sequência e quais ações deve tomar, de forma dinâmica.
- **Continuidade temporal:** Um agente sempre está em um processo contínuo de capacitação.
- **Personalidade:** Característica que facilita a interação com os usuários.
- **Habilidade de comunicação:** Habilidade de interagir com outros agentes ou com usuários para obter informações ou ajudar a cumprir com seus objetivos.
- **Adaptabilidade:** Personalizar preferências do usuário, tendo em vista experiências passadas.
- **Mobilidade:** Possibilidade de se transportar para uma outra máquina, independente da arquitetura e da plataforma.

É difícil encontrar todas estas características em um único agente, mas a maioria deles possui um número substancial das características listadas.

2.3. Tipologia dos agentes

Todos os agentes podem ser classificados em diferentes classes. A tipologia dos agentes tem como objetivo estudar os tipos de entidades, de modo a classificá-los de acordo com suas capacidades, crenças. Segundo (NWANA, 1996), os agentes podem ser classificados nos respectivos tipos:

- **Agentes Colaborativos:** Estes agentes possuem como característica principal a autonomia, habilidade social, reatividade e pro atividade. Para a realização de tarefas, eles cooperam entre si para troca de informações. O objetivo é que um sistema que reúne agentes colaborativos desenvolvidos de formas diferentes permite que o conjunto funcione além das capacidades de um único membro.
- **Agentes de Interface:** As características principais são autonomia e aprendizado. Metaforicamente, estes agentes são assistentes pessoais que colaboram com o usuário dentro do mesmo ambiente.
- **Agentes Móveis:** Possui a habilidade de se transportar de um sistema para outro através da rede, não se restringe ao sistema que inicia sua execução.

Sua vantagem é melhor utilização do tempo e maior transparência na recuperação de informações (ARIDOR; LANGE, 1998).

- Agentes de Informação/Internet: Estes agentes surgiram recentemente, através da necessidade de ferramentas para administrar o crescimento explosivo de informações. Eles desempenham o papel de gerenciar, manipular ou coletar informações de fontes distribuídas.
- Agentes de Software Reativos: Esta tipologia corresponde a uma categoria específica, aos quais os agentes não possuem modelos simbólicos de seu ambiente. Ao invés disso, eles agem e respondem aos estímulos do presente estado do ambiente em que estão.
- Agentes Híbridos: Todas as tipologias citadas acima possuem seus pontos fortes e fracos. Os agentes híbridos tentam maximizar os pontos fortes e minimizar os pontos fracos para um determinado propósito. Eles são o resultado da combinação de duas ou mais tipologias de agentes.

2.4. Sistemas Multiagentes

Um SMA é considerado uma extensão da tecnologia de agentes, onde um grupo de agentes autônomos atua em um ambiente para atingir um objetivo comum. Este objeto pode ser atingido tanto por cooperação, quanto por competição, compartilhando ou não seu conhecimento (BALAJI; SRINIVASAN, 2010).

Existem características fundamentais, as quais distinguem um SMA de sistemas com um único agente (VLASSIS, 2007). Estas características são apresentadas abaixo:

- Design do agente: É comum encontrar SMA em que os agentes são projetados em diferentes formas. Os agentes podem ser baseados em diferentes hardwares ou softwares. Os agentes que possuem comportamentos diferentes entre si são chamados heterogêneos, enquanto aqueles que são projetados de forma idêntica e possuem as mesmas características são chamados de homogêneos. A heterogeneidade dos agentes pode ou não afetar os aspectos fundamentais para a tomada de decisão.

- Ambiente: Agentes devem lidar com ambientes que podem ser ora estáticos, ora dinâmicos. A maioria das técnicas para desenvolvimento com um único agente são feitas para ambientes estáticos, pois é mais fácil de manusear. Em SMA, a presença de múltiplos agentes torna o ambiente dinâmico a partir do ponto de vista de cada agente.
- Percepção: As informações percebidas pelos sensores dos agentes em SMA são tipicamente distribuídas: os agentes podem observar dados que diferem no espaço, aparecem em diferentes localizações; ou temporariamente, percebem em tempos diferentes; ou semanticamente, requer diferentes interpretações. O fato dos agentes poderem observar coisas diferentes torna o ambiente parcialmente observável.
- Controle: Diferente dos sistemas com um único agente, o controle de um SMA é tipicamente descentralizado. Isto significa que grande parte da tomada de decisão encontra-se dentro do próprio agente. Desta forma, é preferível ter um controle descentralizado, ao invés de um controle centralizado por questões de robustez e tolerância a falhas.
- Conhecimento: Em sistemas com um único agente é assumido que o agente conhece todas suas ações, porém suas ações não necessariamente afetam o ambiente. Em SMA, o nível de conhecimento de cada agente sobre o ambiente atual pode variar substancialmente. Em geral, em SMA cada agente deve considerar o conhecimento de todos os outros agentes para a tomada de decisão.
- Comunicação: Interação normalmente é associada a uma forma de comunicação. Em geral a comunicação em SMA é vista como um processo em duas partes, onde todos os agentes podem receber ou enviar mensagens. A comunicação pode ser utilizada para diferentes casos, como: coordenação entre agentes cooperativos, negociação entre agentes. Por isso, é preciso utilizar protocolos para trocar informações de forma segura e rápida, e definir a linguagem que os agentes irão falar para compreensão entre eles (principalmente se eles forem heterogêneos).

2.5. Desenvolvimento

2.5.1. Padronização FIPA

A FIPA (Foundation for Intelligent Physical Agents) é uma associação internacional, sem fins lucrativos, criada em 1996 com o objetivo de desenvolver um conjunto de padrões para a tecnologia de agentes de software. Inicialmente, seus membros eram acadêmicos e organizações industriais, os quais acreditavam que a tecnologia de agentes seria a base de uma nova indústria, capaz de ser utilizados em inúmeras aplicações (BELLIFEMINE; CAIRE, G; GREENWOOD, 2007).

A FIPA possui os seguintes princípios (BELLIFEMINE; CAIRE, G; GREENWOOD, 2007):

- A tecnologia de agentes fornece um novo paradigma para resolver problemas novos e antigos;
- Algumas tecnologias de agente alcançaram um nível de maturidade considerável;
- Para a tecnologia de agentes ser utilizada, é necessário uma padronização;
- A padronização de tecnologias genéricas mostrou resultados eficientes;
- O objetivo é oferecer a padronização de uma infraestrutura e linguagem obrigatórias para a interoperação.

As principais realizações da FIPA foram: o conjunto de especificações padrão que suporta a comunicação interagentes, a linguagem de comunicação entre agentes (ACL) que é bastante utilizada e especificada, um conjunto de protocolos de interação desde uma simples mensagem até a troca de transações complexas.

2.5.2. Plataforma de Agentes - JADE

A plataforma de agentes corresponde a um ambiente onde os agentes são desenvolvidos e executados, podendo ter infraestrutura distribuída ou não. A sua principal responsabilidade é fornecer os principais serviços aos agentes, como: mobilidade, comunicação, controle transacional.

A plataforma escolhida para o desenvolvimento foi a plataforma JADE, por ter código aberto, ser fácil de utilizar e estar complacente com a padronização FIPA.

JADE (Java Agent Development Framework) é um framework criado para facilitar o desenvolvimento de SMA utilizando a linguagem Java. Seu objetivo é criar uma plataforma de agentes para simplificar o desenvolvimento de aplicações, considerando todas as especificações da FIPA para interoperabilidade (BELLIFEMINE; CAIRE, G; GREENWOOD, 2007).

Dentre as especificações, é descrito um modelo de referência da plataforma, o qual identifica os agentes necessários para gerenciar a plataforma: o agente de gerenciamento do sistema (AMS), responsável por manter um diretório dos agentes residentes na plataforma e gerir seus ciclos de vida; o agente de comunicação canal (ACC), que fornece o caminho para o contato entre agentes dentro ou fora da plataforma; e o facilitador diretório (DF), que fornece serviços de páginas amarelas para a plataforma de agentes (BELLIFEMINE; CAIRE, G; GREENWOOD, 2007).

Ao criar uma plataforma utilizando JADE, automaticamente os agentes AMS, DF e ACC são ativados. Todos os serviços oferecidos por eles são disponíveis através de métodos na linguagem Java, abstraindo a complexidade.

As especificações também definem uma linguagem de comunicação entre agentes (ACL), utilizada para trocar mensagens entre os agentes. Um dos grandes diferenciais de JADE é o suporte a todo tipo de comunicação através de diversos serviços como: serviços de nomes e páginas amarelas, codificação e decodificação de mensagens, transporte de mensagens e bibliotecas com protocolos de interação.

3. Trabalhos Relacionados

3.1. Introdução

A comparação de compras na internet é o processo de coletar informações de um determinado produto em diferentes sites de compra. Existe um grande desafio por trás desta comparação, pois cada site possui sua própria maneira de organizar componentes e ofertas.

Este capítulo tem como objetivo apresentar o estado da arte, através de bibliografias que utilizam agentes inteligentes para realizar esta comparação. Os trabalhos mostram inúmeras formas de estruturar SMA e como os agentes podem ser utilizados para extrair informações de produtos, servindo como apoio a parte prática desta dissertação.

3.2. ShopBot [DOORENBOS; ETZIONI; WELD, 1997]

ShopBot (DOORENBOS; ETZIONI; WELD, 1997) é um sistema capaz de analisar a descrição de produtos e seus atributos em lojas online, utilizando a combinação de heurísticas de busca, reconhecimento de padrões e técnicas de aprendizagem indutivas.

O sistema opera em duas fases: a fase de aprendizagem, um aprendiz off-line responsável por criar uma descrição do vendedor; e a fase de comparação de lojas, um comprador em tempo real que utiliza as descrições do produto para ajudar o usuário a escolher o produto com menor preço.

A fase de aprendizagem é feita através de um agente aprendiz. O objetivo é analisar cada site de venda para aprender a descrição simbólica de cada um deles, a fim de construir uma forma de extrair a informação do produto. Esta fase pode ser feita off-line e precisa ser feita uma vez em cada site.

O ShopBot foi projetado para explorar as regularidades dos compradores, para facilitar a compra de usuários humanos. Estas regularidades são consideradas cruciais para agentes em tempo real e dinâmicos, pois eles são projetados para tirar proveito delas. Elas são listadas a seguir.

- Regularidade da navegação: Lojas online são projetadas para achar produtos rapidamente. Esta afirmação sugere que as lojas possuam um campo para pesquisa na página principal, para facilitar a navegação do usuário.
- Regularidade da uniformidade: Mesmo que as lojas apresentem seus produtos se forma diferente, todas elas descrevem produtos e seus atributos de modo semelhante.
- Regularidade da separação vertical: O espaço em branco, as novas linhas são utilizadas para facilitar a compreensão do cliente para informações diferentes.

As regularidades observadas acima sugerem uma abordagem de aprendizagem para o problema de encontrar a descrição dos produtos em diferentes sites. Baseado na regularidade de uniformidade, assume-se que as descrições do produto respeitam um mesmo formato até um certo nível de abstração. O algoritmo de aprendizagem deve induzir o número de páginas, as descrições dos produtos.

Existem três componentes necessários para o algoritmo de aprendizagem: identificação de um formulário de busca, preenchimento do formulário e descoberta do formato da descrição dos produtos na página de resultados da busca. Primeiramente, o agente aprendiz deve achar potenciais formulários de busca na página inicial do site que recebeu como entrada. São utilizadas heurísticas simples para descartar os formulários que não estão claros e escolher apenas o formulário de busca.

O modelo de domínio inclui algumas pesquisas para teste. Desta forma, o segundo passo inclui a realização dos testes no formulário de busca. O terceiro passo é a tarefa mais complexa, onde serão analisados os resultados que retornam do formulário. Os testes incluem dois tipos: páginas “erradas”, onde não existe nenhum produto que combina com o parâmetro de busca; e páginas de “sucesso”, onde apresentam produtos populares e um ou mais itens combinam com o parâmetro de busca.

O primeiro teste serve para o agente descobrir qual é a reação do site de compras ao se deparar com uma palavra estranha. Este é o conhecimento necessário para o agente saber determinar se a busca foi bem sucedida ou não. As páginas de resultados diferentes das páginas “erradas” são consideradas sucesso na pesquisa.

O segundo teste serve para fazer buscas em produtos que são populares, pois existe a garantia de resultados na pesquisa. Através da página de resultados, o agente deve induzir o formato da descrição dos produtos. São utilizadas linguagens de abstração para o código HTML da página para descobrir formulários padrão. Um exemplo utilizando o código HTML “Click Aqui <a href=’http://www.google.com’ para pesquisar.”, ao aplicar a abstração o código seria se transforma em: “texto<a>textotexto.”.

Após a abstração, aplicando a regularidade de separação vertical, o corpo da página é quebrado em vários trechos. O agente assume que toda a descrição do produto começa em uma nova linha, ou seja, em um trecho do corpo da página que foi quebrado. São analisados cada trecho procurando por um padrão. Ao definir um padrão, o agente deve armazenar em seu conhecimento, para saber a forma de retirar do texto as informações de cada produto.

A fase de comparação de lojas é feita após o agente recolher a informação no resultado da busca sobre os produtos e seus atributos. Para o respectivo trabalho, a comparação é feita apenas nos preços do produto, ordenando-os de forma ascendente. Para isso, é necessário comparar apenas o atributo dos produtos correspondente ao preço e mostrar ao usuário final o resultado da ordenação.

Os experimentos demonstram que o ShopBot é um agente que navega com sucesso em diferentes lojas e consegue extrair apenas as informações relevantes. Ele aprende automaticamente como comprar nas lojas online e requer apenas um conhecimento mínimo sobre diferentes domínios de produto.

Um dos pontos fracos deste sistema é a limitação apenas em lojas que oferecem campo de busca. Algumas lojas, principalmente lojas pequenas, não oferecem este recurso e a única opção de navegação é através da hierarquia. Existe a necessidade de o ShopBot operar sob essas hierarquias. Outro ponto fraco é a

restrição às páginas no formato HTML, excluindo as páginas feitas com outras tecnologias, como java script.

3.3. ShopBot da Universidade de Hanyang [CHOI; HAM; KIM; LEE; YANG, 2000]

Membros da Universidade de Hanyang apresentam o ShopBot (CHOI; HAM; KIM; LEE; YANG, 2000), um agente de comparação de compras. Grande parte deste sistema foi baseado no ShopBot (seção 3.2), mas apresenta diferenças significativas de grande importância para este trabalho, como: o maior detalhamento da extração das informações, a generalização de regras de extração de informação para qualquer organização de página.

O sistema é composto por: um algoritmo de aprendizado, que determina a posição do produto no código HTML da página de busca; e um método para generalização do produto, que é aplicado aos resultados da pesquisa para reconhecer palavras diferentes com significados iguais.

O algoritmo de aprendizagem é utilizado pelo agente para criar uma regra de extração de informações específica para cada loja online. Como entrada, o agente recebe uma URL para aprender a localização das unidades de descrição do produto (PDU). O agente traduz o dado da localização em um wrapper. Cada site deve ter um wrapper correspondente para a extração da informação de seus produtos. Ele é uma estrutura que compreende a localização da informação e a traduz de modo que os agentes possam utilizar para a extração das informações do produto da respectiva fonte. Para este algoritmo, assume-se que uma mesma loja disponibiliza as informações de todos seus produtos de forma igual ou similar.

Para extrair as PDU, são considerados dois tipos de classificação para os resultados de busca: formulários fixos, como por exemplo tabelas, e formulários semiestruturados. O processo de aprendizagem para gerar o wrapper é feito separadamente para cada tipo.

Em geral, os resultados de pesquisa são do tipo tabela, onde a primeira linha contém os títulos referentes aos tipos de informação de cada coluna, e as outras linhas correspondem ao PDU de cada produto fornecido pelo site de compras. O aprendizado do texto é feito pelo agente aprendiz, que utiliza os termos da primeira

linha como base para compara os termos com a ontologia e define o significado correspondente para cada coluna.

Para lidar com casos mais genéricos, existe o algoritmo de aprendizagem para formulários semiestruturados. Em geral, o formato esperado pelas páginas é a seguinte: cabeçalho, uma lista de produtos e um rodapé.

Para fazer a análise da página, o cabeçalho e o rodapé são ignorados, pois são irrelevantes para a extração de informações. O algoritmo de aprendizagem deve identificar a posição dos PDU no corpo da página seguindo quatro passos. Primeiramente, o agente deve salvar a página do resultado da busca em um arquivo. Logo após, este arquivo é quebrado em algumas linhas lógicas. O algoritmo examina as tags delimitadoras do HTML, como por exemplo
. Esta linha lógica é conceitualmente similar à linha que o usuário vê no browser. O algoritmo deve analisar as linhas lógicas para reconhecer os seus significados, que serão expressos em números que identificam o conteúdo da linha. Por fim, toda a página que está expressa em uma sequência de números deve ser examinada a fim de procurar um padrão repetido. O padrão repetido é extraído pelo agente, tornando-se uma estrutura PDU.

O método para generalização do produto é utilizado pelo agente para conhecer terminologia ou palavras-chave para descrever o produto, compreendidos em ontologias. Este é utilizado para reconhecer o tipo de informação de qualquer loja online. Se um site em particular utiliza a palavra “nosso preço”, o agente deve reconhecer esta informação como o atributo preço.

A proposta do método é construir gradativamente ontologias de forma automática. A Figura 3.1 ilustra este mecanismo por completo. Primeiramente, o agente de interface recebe as informações de busca. A estrutura de análise é responsável pela criação das regras de extração para as respectivas fontes de informação. Nos casos onde as regras de extração foram geradas com sucesso, o gerador de ontologias envia várias buscas para o shopping, extraindo os novos termos dos resultados da busca e adicionando eles às ontologias já existentes no conhecimento do domínio.

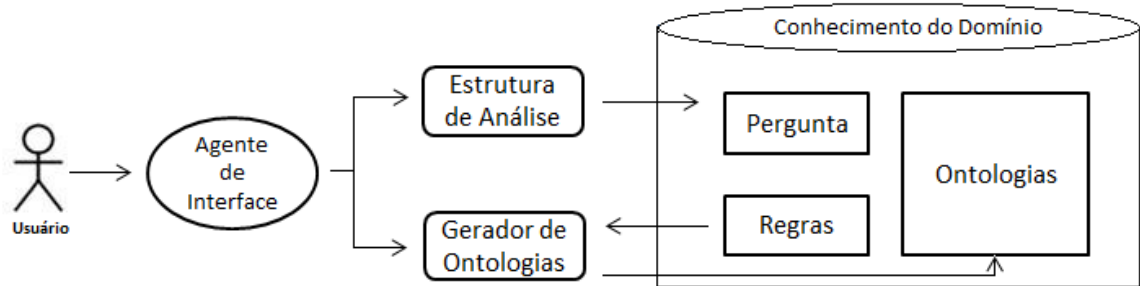


Figura 3.1: O gerador de ontologias (CHOI; HAM; KIM; LEE; YANG, 2000)

As ontologias são organizadas em uma estrutura hierárquica. Esta estrutura tem como objetivo ilustrar os termos e suas relações. O conceito corresponde a produtos, ou palavras que possam identificar o mesmo, já as terminologias correspondem aos atributos de descrição do produto. É possível existir relacionamento entre conceitos do tipo IS-A ou HAS-A. O relacionamento IS-A é um conjunto de termos utilizados como sinônimo do conceito, e o relacionamento HAS-A é um conjunto de termos que expressam características do conceito. A Figura 3.2 mostra um conjunto de conceitos e terminologias, os quais possuem ambos os relacionamentos.

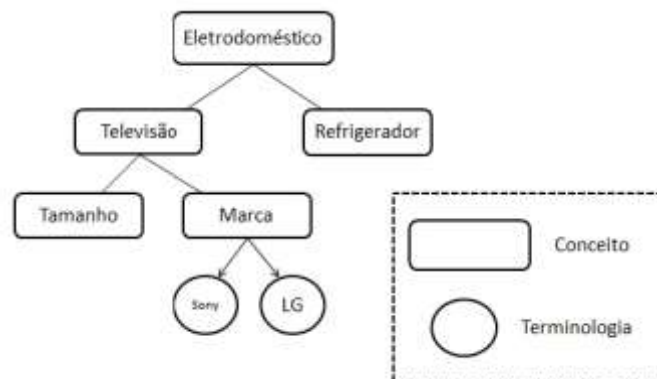


Figura 3.2: Estrutura das ontologias (CHOI; HAM; KIM; LEE; YANG, 2000)

Os experimentos feitos para avaliar o desempenho deste sistema composto pelo algoritmo de aprendizado e pelo método para generalização do produto foram muito positivas. Em 94% dos testes, o algoritmo de aprendizado conseguiu reconhecer e criar o wrapper para os respectivos sites. O ponto fraco deste sistema é a restrição às páginas no formato HTML, excluindo as páginas feitas com outras

tecnologias, como java script. Outro ponto fraco é a dificuldade em extrair as informações do produto em formulários semiestruturados sem ter uma ontologia apropriada.

A proposta do sistema é criar um novo mecanismo para extrair a descrição do produto, além de um mecanismo automático para a criação de ontologias. Um dos pontos positivos é o poder de escalabilidade e a habilidade de aprendizado dinâmico ao reconhecer a localização da descrição do produto, sem ter que remover fragmentos inúteis diversas vezes. Por consequência, o sistema possui melhor desempenho se comparado aos sistemas que procuram por toda a página os produtos nas lojas online a cada busca nova.

3.4. Sistema de Compras Inteligentes [MENG; ZENG, 2005]

O Sistema de Compras Inteligentes foi proposto por membros da Universidade de Wuhan, China (MENG; ZENG, 2005). O principal objetivo é utilizar tecnologias de agentes para automatizar o processo de compras, adaptando as opções de acordo com as preferências do usuário. O grande diferencial deste sistema é que sua comparação entre produtos não leva em consideração apenas o melhor preço, mas também a qualidade, a marca do produto tendo em vista o perfil do usuário.

O objetivo do sistema é automatizar o processo de compras, auxiliando os clientes a ter as informações de mercadoria em diferentes lojas, mostrando uma comparação dos produtos disponíveis na internet. Para automatizar o processo de compras, o sistema deve possuir as seguintes características:

- **Autonomia:** O sistema deve gerar uma lista de mercadorias e juntar com as informações da mercadoria de diversos sites de compra. O sistema deve ter um mecanismo de comparação para ajudar o consumidor a escolher a melhor opção.
- **Adaptação as preferencias do consumidor:** O sistema deve aprender a preferencia individual de cada cliente, ao analisar o comportamento recente do consumidor e os produtos que foram comprados.

- Acesso a múltiplas lojas online: O sistema de compras deve reunir, de forma automática, as informações de mercadorias de diversas lojas.

Para contemplar todas as características, foi adotado SMA com trabalho colaborativo entre eles. Ao todo, o sistema é composto por cinco tipos de agentes que interagem entre si: agente da interface, gerente de compras, agente de compras, agente de avaliação e agente de preferência. A interação é feita através da troca de mensagens entre os agentes. A estrutura do sistema é ilustrada na Figura 3.3.

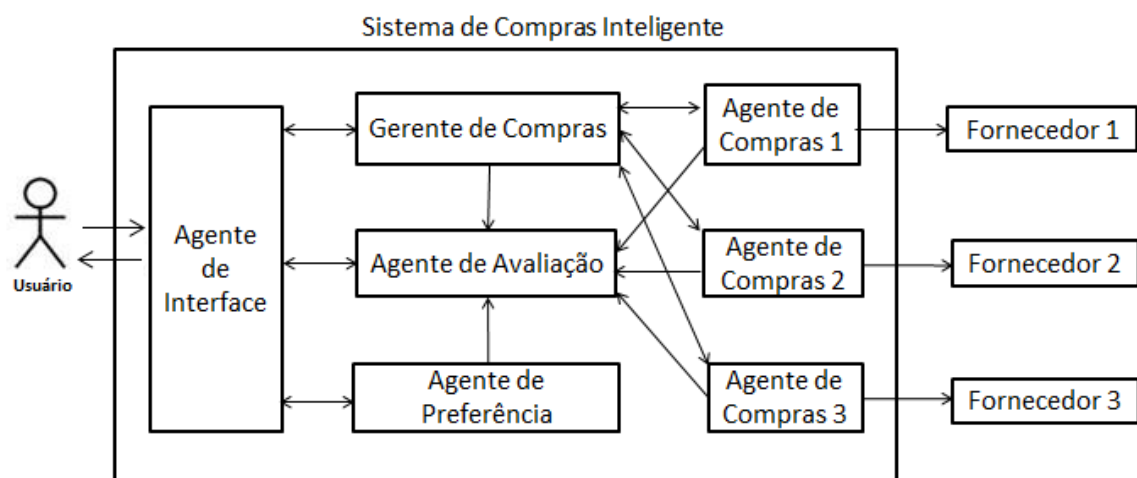


Figura 3.3: A estrutura do sistema multiagente (MENG; ZENG, 2005)

O principal trabalho do agente de interface é realizar a comunicação bidimensional entre o usuário, que também pode ser considerado cliente, e o sistema de compras. Quando o cliente digita a mercadoria desejada, o agente de interface envia para os outros agentes do sistema esta informação. Após o processo de compra ser concluído, o resultado é enviado para a interface através deste mesmo agente.

O gerente de compras responde a busca do cliente. Ele é responsável por gerar uma lista de mercadorias e buscar as informações sobre as mesmas em diversos sites de compras. Para as buscas serem mais eficientes é criado um grupo de agentes compradores, onde cada um realiza a busca da mercadoria em diferentes sites de compras. A busca é feita em paralelo e o gerente de compras

deve gerenciar o estado de execução de cada agente comprador e coordenar a distribuição de tarefas entre eles.

Quando um agente comprador é criado, eles são enviados para procurar informações de mercadorias de um respectivo site de compra. A informação segue um formato definido, que contém: a categoria da mercadoria que o cliente quer comprar, e uma lista de objetos contendo respectivamente o produto oferecido pelo fornecedor e o fornecedor. Eles retornam o resultado da busca para o agente de avaliação.

Após a coleta das informações pelos agentes compradores, o resultado da busca é passado ao agente de avaliação. Este agente é muito importante para o sistema de compras online, pois realiza a comparação entre produtos, avaliando cada mercadoria para então selecionar a melhor opção. Esta opção é mostrada ao usuário e ele deve definir se está satisfeito ou não com o resultado da busca.

Caso o cliente não esteja satisfeito com a mercadoria que foi recomendada, o agente de preferência vai adquirir a preferência do cliente em sua base de conhecimento e fazer outra análise baseada em preferências. Um jeito simples de adquirir a preferência do usuário é oferecer uma interface onde ele possa interagir sobre uma lista e selecionar a sua preferência. Outra alternativa é aprender a preferência do cliente através de seu histórico de compras.

A proposta do sistema de comparação de compras baseados em agentes reduz o tempo e o esforço do cliente para compras, oferecendo uma lista de mercadorias que reúne informações de diferentes sites de compra. O sistema adota tecnologias que podem ser reutilizadas, pois os agentes são projetados como componentes de software independentes entre si.

3.5. Assistente Pessoal de Viagens [ALHASHEL; BALACHANDRAN; MOHAMMEDIAN, 2009]

O Assistente Pessoal de Viagens é um SMA proposto por (ALHASHEL; BALACHANDRAN; MOHAMMEDIAN, 2009) com foco específico para a indústria de viagens. Clientes e agências de viagens podem ser beneficiadas através do uso

deste sistema, pois ele oferece pacotes de viagens de acordo com as necessidades do cliente.

A Figura 3.4 ilustra o modelo de negociação baseado em agentes para resolver o problema do assistente de viagens. Neste modelo, as negociações são feitas através de um agente moderador que atua como o único intermediário entre o agente cliente e os fornecedores de serviços.

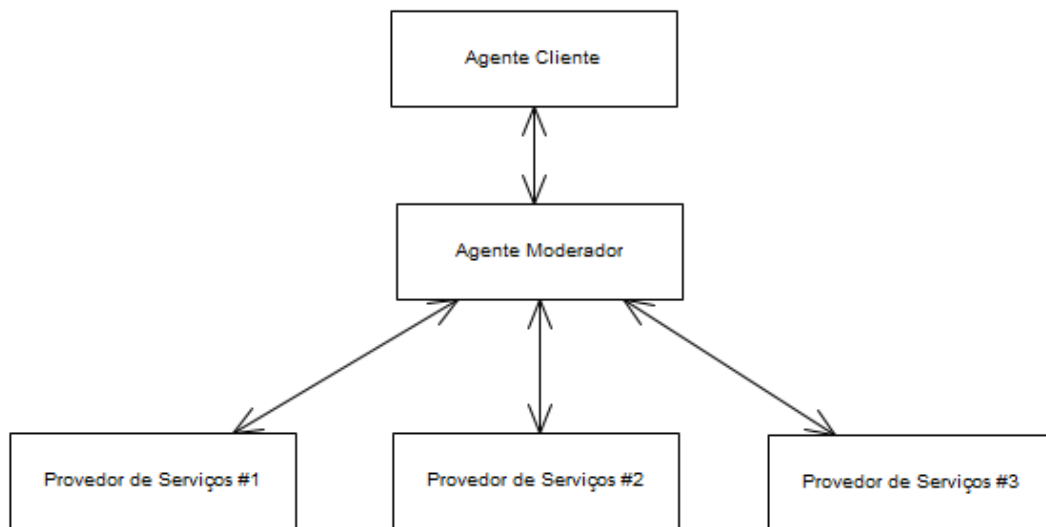


Figura 3.4: Negociação multiservidor com um moderador (ALHASHEL; BALACHANDRAN; MOHAMMEDIAN, 2009)

O agente moderador é responsável por coordenar um conjunto de provedores de serviços para a composição do pacote. Ele repassa aos provedores de serviços as informações repassadas pelo usuário, e cada provedor responde se é possível ou não atender às necessidades do cliente. Caso não seja possível, o agente mediador deve recorrer a outros provedores, até que seja feito um pacote de viagens com os critérios definidos pelo cliente.

A arquitetura multiagente da aplicação é composta por cinco agentes: agente consumidor, agente moderador, agente reserva hotel, agente reserva voo, agente aluga carro.

O agente consumidor fornece ao usuário uma interface gráfica que permite estabelecer critérios para o pacote de viagem, como por exemplo: preço máximo, preço mínimo. Este agente se comunica com o agente moderador para obter um pacote de viagens que atenda as solicitações do cliente.

O Agente Moderador é basicamente o papel de um assistente de viagens. Ele tenta criar um pacote de viagens de acordo com os critérios definidos, maximizando a utilidade do pacote de viagens sobre estes critérios. Caso um pacote de viagens que atenda aos critérios seja encontrado, o agente moderador agenda e realiza os seguintes pagamentos: aluguel de carro, quarto de hotel e voo.

Os agentes reserva hotel, reserva voo e aluga carro possuem os mesmos comportamentos, porém em diferentes serviços. Cada um desses agentes tem autonomia para fazer reservas caso o agente moderador aceite a proposta enviada por eles. Para cada reserva concluída com sucesso, as informações detalhadas de cada serviço são repassadas ao agente moderador.

O sistema foi desenvolvido utilizando o framework JADE. Este framework foi escolhido por estar de acordo com as especificações FIPA. A comunicação entre agentes é uma das maiores características disponíveis pelo framework. Os maiores componentes do sistema são ilustrados pela Figura 3.5.

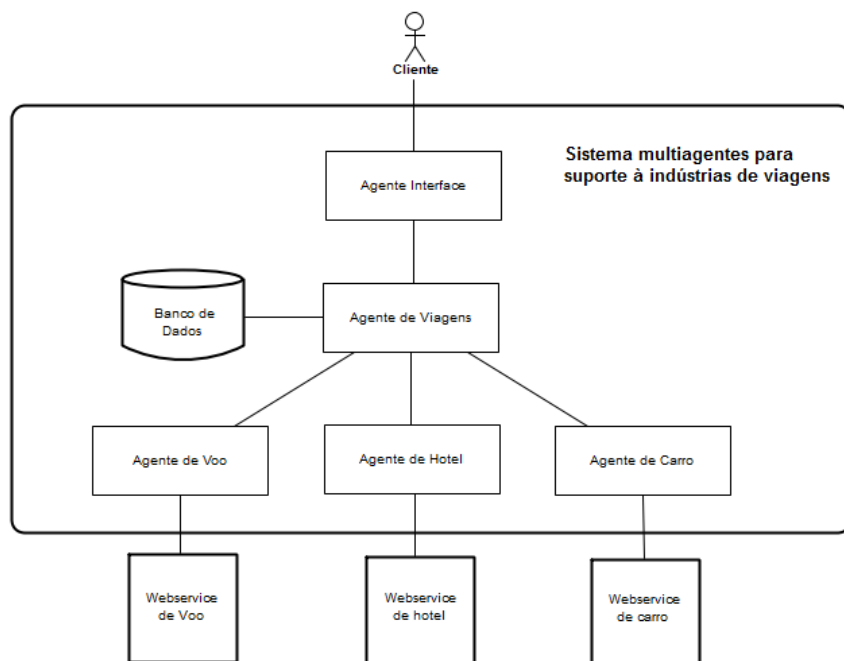


Figura 3.5: Componentes do sistema Assistente Pessoal de Viagens
(ALHASHEL; BALACHANDRAN; MOHAMMEDIAN, 2009)

O cliente acessa o sistema e fornece critérios de busca através da interface do usuário. Esta interface também recebe as informações dos pacotes sugeridos

pelo sistema, possibilitando ao usuário optar e agendar pacotes, além de mostrar as informações detalhadas do pacote após a conclusão do agendamento.

A base de dados contém as informações de viagens, tais como: voos, acomodações e carros, como também os detalhes do cliente.

O sistema possui cinco tipos de agentes para atingir o objetivo e agendar os pacotes apropriados de viagem. O agente de viagens corresponde ao agente moderador e é responsável por procurar o pacote de viagens utilizando os critérios de busca do cliente e incorpora com os outros agentes para procurar a disponibilidade de agendamento e agendar o pacote de viagens. Os agentes de Voo, Hotel e Carro interagem com webservices para receber informações sobre disponibilidade de agendamento e criar o pacote. Esses agentes são ativados pela ação do o agente moderador.

Este artigo apresenta uma abordagem colaborativa baseada em agentes para desenvolver um assistente pessoal de viagens. O agente moderador é descrito como coordenador e negociador para resolver o problema. Os requisitos do cliente são alcançados através de uma série de negociações entre o agente moderador e os provedores de serviço. Este trabalho demonstra o quão promissor pode ser a tecnologia de agentes para problemas de negócio real.

3.6. Considerações Finais sobre as Abordagens Envolvendo SMA na Busca de Informações em Sites de Compra

Neste capítulo, foram apresentadas abordagens que mostram SMA na busca de informações em sites de compra. O objetivo do estudo serviu como base para o desenvolvimento deste trabalho.

Por exemplo, o ShopBot [DOORENBOS; ETZIONI; WELD, 1997] propõe uma fase inicial de aprendizagem off-line, com o objetivo de analisar cada site de venda para extrair a informação do produto.

No ShopBot da Universidade de Hanyang [CHOI; HAM; KIM; LEE; YANG, 2000], a extração da informação do produto é feita através de um wrapper. O wrapper tem como objetivo mapear a localização as informações de um produto.

Cada agente é responsável por receber uma URL e criar um wrapper correspondente.

O Sistema de Compras Inteligentes [MENG; ZENG, 2005] propõe um SMA com trabalho colaborativo entre eles. O agente de interface realiza a comunicação bidimensional entre o usuário. O gerente de compras é responsável por gerar uma lista de mercadorias e buscar as informações sobre as mesmas em diversos sites de compras. A busca é feita em paralelo e o gerente de compras deve gerenciar o estado de execução de cada agente comprador e coordenar a distribuição de tarefas entre eles.

Já no Assistente Pessoal de Viagens [ALHASHEL; BALACHANDRAN; MOHAMMEDIAN, 2009], a busca das informações é feita através de provedores de serviços. O agente moderador é responsável por repassar a busca do usuário aos provedores de serviços, os quais retornam as informações desejadas ao agente.

3.7. Conclusão

Este capítulo teve por objetivo discutir propostas que utilizem SMA para a extração de informações do comércio eletrônico, comprovando o expressivo interesse nessa área.

Com relação à estrutura de SMA e a utilização de agentes para coletar informações de produtos em lojas virtuais, objetivo de estudo deste trabalho, algumas abordagens foram estudadas com o propósito de servir como inspiração ao desenvolvimento do SMA proposto.

4. Sistema BuscaCompra

4.1. Introdução

Para (WOOLDRIGE; JENNINGS, 1995), os SMA oferecem diversas vantagens se comparados aos sistemas convencionais, dentre eles: resolução mais rápida de problemas utilizando paralelismo, redução da quantidade de dados através das soluções parciais transmitidas entre agentes, maior flexibilidade na resolução de problemas, pois os agentes utilizam diferentes habilidades.

Tendo em vista estas diversas vantagens, optou-se por fazer um SMA para a coleta de informações de produtos em diferentes sites de compra. Este capítulo tem como objetivo apresentar todo o sistema, através de requisitos, modelagem e arquitetura. Por fim, é apresentado o desenvolvimento do sistema, dando ênfase às tarefas de cada agente.

4.2. Apresentação do Sistema

Existem muitos trabalhos na literatura que utilizam agentes inteligentes na busca de informação distribuída (DOORENBOS; ETZIONI; WELD, 1997; CHOI; HAM; KIM; LEE; YANG, 2000; MENG; ZENG, 2005). O sistema BuscaCompra é um sistema multiagente que foi desenvolvido para buscar informações distribuídas sobre produtos em diferentes sites de compra disponíveis na internet.

O objetivo é permitir que o usuário faça pesquisas de um determinado produto e o sistema ofereça informações de diferentes sites de compra. Desta forma, ele possui a liberdade para comparar os produtos de diversos sites e fazer uma compra mais rápida e criteriosa.

A pesquisa aos diversos sites de compra é realizada no momento em que o usuário está consultando o site, através de agentes inteligentes que são criados no momento em que a busca é solicitada. Desta forma, é possível associar a utilização dos agentes na aplicação para substituir o armazenamento de dados físico.

Os sites de compra que são realizados a pesquisa é o grande diferencial deste sistema, pois refletem o cenário atual do comércio eletrônico. Este cenário é composto por dois modelos de negócio: lojas virtuais e compras coletivas. Atualmente, não existe nenhuma ferramenta que tenha como resultado de busca a composição destes dois modelos.

O conceito de loja virtual é utilizado para todos os sites que possuem produtos a venda na internet, enquanto o conceito de sites de compra coletiva é caracterizado por oferecer descontos em serviços e produtos a um grupo de compradores.

4.3. Requisitos

Os requisitos do sistema foram divididos em funcionais e não funcionais. Para o desenvolvimento do sistema os seguintes requisitos foram considerados:

4.3.1. Requisitos Não Funcionais

1. Requisito não funcional 1 – Desenvolvimento do sistema: O sistema deve ser uma aplicação Web desenvolvida na linguagem Java, versão 6, utilizando o servidor de aplicação JBOSS versão 7.0.2 e a ferramenta de gerenciamento de dependências Maven, versão 2.1.1.

2. Requisito não funcional 2 – Desenvolvimento de agentes: Os agentes inteligentes serão implementados utilizando a versão 4.2.0 do framework JADE.

3. Requisito não funcional 3 – Manipulação HTML: As manipulações de tags HTML serão feitas através da utilização da biblioteca Jsoup, versão 1.7.1.

4. Requisito não funcional 4 – Desenvolvimento da interface gráfica: A interface gráfica deve ser desenvolvida utilizando a biblioteca Primefaces, versão 3.4.1.

5. Requisito não funcional 5 – Interface gráfica para o usuário: A página inicial do sistema deve conter um campo aberto e um botão de “Buscar”.

6. Requisito não funcional 6 – Resultados da busca: A comunicação entre o sistema e o usuário deve estar clara, de modo que seja comunicado o resultado da busca caso sejam encontrados e caso não tenham resultados deverá mostrar uma mensagem de erro na busca.

4.3.2. Requisitos Funcionais

1. Requisito Funcional 1 – Busca por produtos: O sistema deve permitir que o usuário realize buscas por produto a partir de uma ou mais palavras digitadas na interface gráfica.

2. Requisito Funcional 2 – Realização das buscas: O sistema só irá realizar buscas caso o usuário tenha digitado duas ou mais letras no campo aberto.

3. Requisito Funcional 3 – Busca em sites de compra coletiva: O sistema deve contemplar a busca da palavra-chave em lojas virtuais e em sites de compra coletiva da cidade de Florianópolis, nos seguintes sites de compra coletiva: www.peixurbano.com.br, www.surfandovantagens.com.br, www.loveonline.com.br e www.groupon.com.br.

4. Requisito Funcional 4 – Busca em sites de lojas virtuais: O sistema deve utilizar o webservice disponível pela empresa Bucapé Developer Company para buscar os dados nas seguintes lojas virtuais: Walmart, Zuba, Casas Bahia, Ponto Frio, FastCam, Magazine Luiza, Extra, Fast Shop, dentre outras.

5. Requisito Funcional 5 – Resultado da busca com sucesso: O resultado da busca, se realizada com sucesso, deve retornar informações sobre produtos em diferentes sites de compra. Para este caso, o sistema deverá mostrar as informações dos produtos em forma de lista, informando o nome do produto, o nome e a imagem da loja virtual, valor do produto e um link para a página do produto.

6. Requisito Funcional 6 – Enquanto carrega os dados da busca: Enquanto a busca está sendo feita e os dados estão sendo carregados, o sistema deve impossibilitar que o usuário faça qualquer tipo de interação com a interface.

7. Requisito Funcional 7 – Resultado da busca sem sucesso: Caso a busca não retorne nenhuma informação, o sistema deve apresentar uma mensagem para o usuário avisando que não foi encontrado nenhum produto com aquela especificação.

4.4. Metodologia

Segundo (SOMMERVILLE, 2007), a engenharia de software propõe a utilização de metodologias para estruturar o desenvolvimento de software com o objetivo de facilitar a produção de alta qualidade e rentável. A metodologia é composta por inúmeros componentes como: diagramas e documentos, que representam o desenvolvimento do sistema de forma gráfica e textual.

A área de engenharia de software baseada em agentes é um nova área e pode ser vista como uma evolução da programação orientada a objetos (BELLIFEMINE; CAIRE, G; GREENWOOD, 2007). Apesar da utilização de agentes oferecer eficiência para a resolução de problemas em algumas áreas, existe uma carência em metodologias para o desenvolvimento de software baseado em agentes (CLAIRE; BAHRI; NIKRAZ, 2006).

A modelagem do sistema BuscaCompra foi baseada na metodologia de desenvolvimento de sistemas multiagentes utilizando o framework JADE, proposta por (CLAIRE; BAHRI; NIKRAZ, 2006). Esta metodologia foi criada com o objetivo de formalizar a análise e o planejamento do desenvolvimento de SMA. O seu propósito não é estender as técnicas de orientação a objetos, mas sim se especializar nos agentes e nas suas abstrações.

Esta metodologia é composta por um conjunto de diagramas, cada um ilustrando partes diferentes do sistema. O objetivo é identificar as responsabilidades dos agentes e a comunicação entre eles para modelar o problema e a solução proposta pelo SMA.

O diagrama de casos de uso é uma forma eficiente de identificar os requisitos funcionais de um sistema. O seu objetivo é ilustrar a forma que o sistema deve interagir com o usuário final e como o sistema deve interagir com outros sistemas

para alcançar seu objetivo. A Figura 4.1 ilustra o diagrama de casos de uso do sistema BuscaCompra.

A interação do usuário oferece somente a possibilidade de buscar produtos. Esta busca é composta por duas interações: a primeira é a comunicação com o webservice para lojas virtuais, a segunda é a comunicação com inúmeros sites de compra coletiva através da utilização de crawlers focado. Um crawler focado tem como objetivo acessar uma página da internet e extrair o seu código em formato HTML (LUI, 2011).

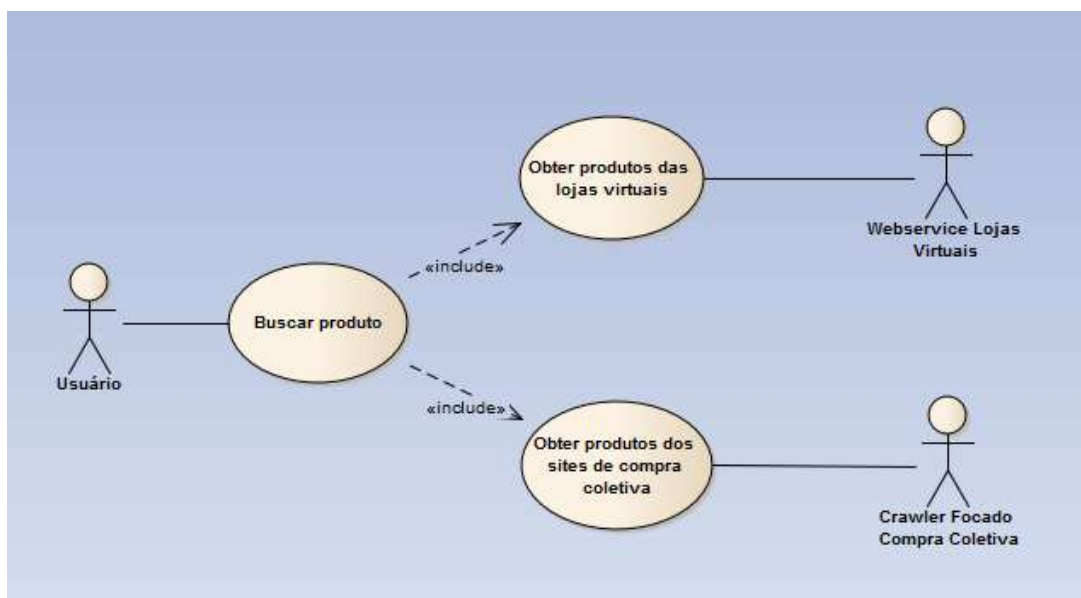


Figura 4.1: Diagrama de casos de uso (CLAIRE; BAHRI; NIKRAZ, 2006)

A identificação dos tipos de agente que vão compor o sistema é feita através do diagrama de agentes. Este diagrama é composto por quatro tipos de elementos: tipos de agentes, que são representados por círculos; humanos, as pessoas que interagem com o sistema e são representados pelo símbolo de ator da UML; recursos, os sistemas externos que devem interagir com o sistema, representados por retângulos ; e os conhecidos, representados pela linha que liga todos os elementos descritos acima, especificando a interação entre eles (CLAIRE; BAHRI; NIKRAZ, 2006).

A Figura 4.2 apresenta o diagrama de agentes do sistema BuscaCompra. Os tipos de agentes presentes no sistema são: Agente Interface, Agente Sistema, Agente Loja e Agente Compra Coletiva.

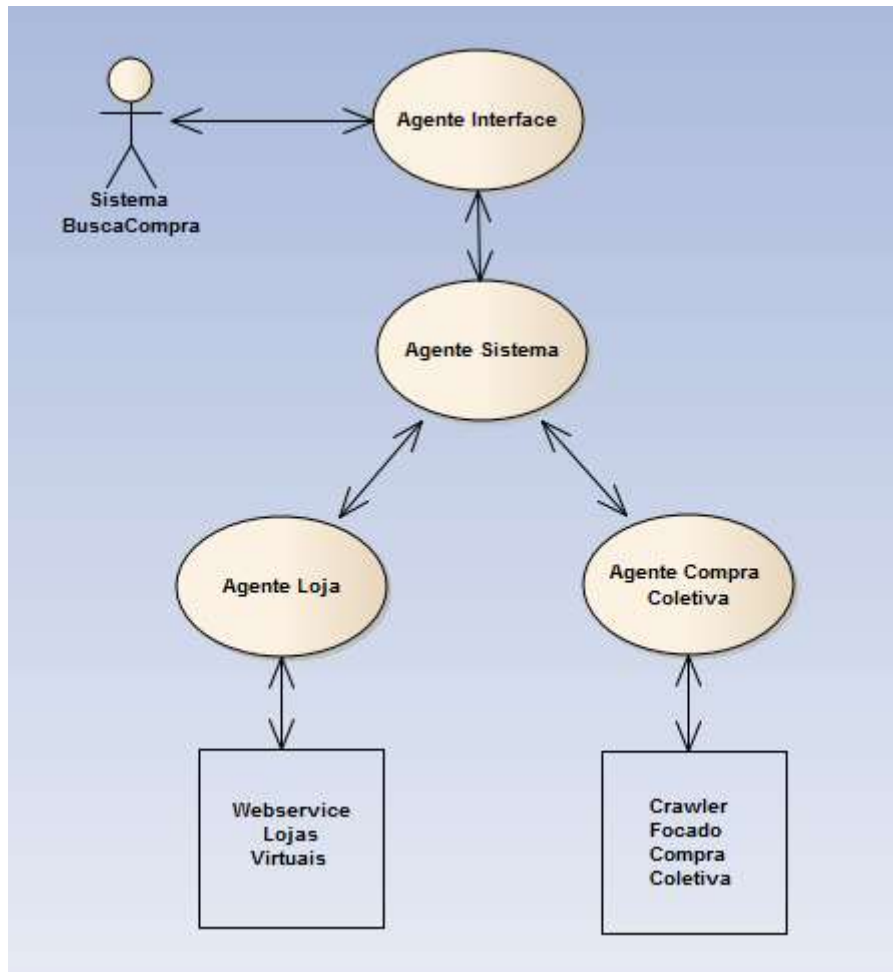


Figura 4.2: Diagrama de agentes (CLAIRE; BAHRI; NIKRAZ, 2006)

A interação com o agente interface deve ser através do próprio sistema BuscaCompra. O usuário do sistema interage com a camada visual e, após as validações necessárias descritas nos requisitos funcionais, o próprio sistema cria e se comunica com o agente interface.

O sistema não possui nenhum tipo de armazenamento de dados. Todas as informações obtidas da busca do usuário devem ser feitas em tempo de execução do programa, e por este motivo, a aplicação utiliza recursos externos como fonte de dados. Conforme o diagrama de agentes apresenta, o agente loja se comunica com um webservice para coletar dados de lojas virtuais, enquanto o agente compra coletiva se comunica através de um crawler focado para coletar dados dos sites de compra coletiva.

A metodologia de (CLAIRE; BAHRI; NIKRAZ, 2006), além de diagramas, propõe a identificação de cada tipo de agente através de uma lista de

responsabilidades feita de forma informal e intuitiva, chamada de tabela de responsabilidades. O principal objetivo desta tabela é determinar a responsabilidade de cada agente, abstraindo a comunicação entre eles. A Tabela 4.1 apresenta as responsabilidades dos agentes que compõe o sistema BuscaCompra.

Tabela 4.1: Tabela de Responsabilidades (CLAIRE; BAHRI; NIKRAZ, 2006)

Tipo de Agente	Responsabilidades
Agente Interface	<ul style="list-style-type: none"> • Informar a palavra-chave informada pelo usuário • Receber a lista de produtos encontrados nos sites
Agente Sistema	<ul style="list-style-type: none"> • Criar um agente loja para buscar produtos de lojas virtuais • Receber a lista de produtos de lojas virtuais • Possuir uma lista de sites de compra coletivas cadastrados no sistema • Criar um agente compra coletiva para cada site cadastrado no sistema • Receber a lista de produtos de sites de compra coletiva • Reunir informações provenientes do agente loja e dos agentes compra coletiva
Agente Loja	<ul style="list-style-type: none"> • Buscar produtos de lojas virtuais que tenham a palavra-chave • Retornar a lista de produtos encontrada
Agente Compra Coletiva	<ul style="list-style-type: none"> • Buscar produtos de sites de compra coletiva que tenham a palavra-chave • Retornar a lista de produtos encontrada

4.5. Estrutura dos Agentes

O sistema BuscaCompra possui coleção de agentes autônomos, que colaboram entre si, com o objetivo de realizar a busca em diferentes sites de venda. A Figura 4.3 apresenta a estrutura dos agentes: agente interface, agente sistema, agente loja e agente compras coletivas. A seta bidirecional representa a comunicação entre eles.

Para desenvolver um sistema multiagentes utilizando o framework JADE é preciso conhecer as características básicas que o framework oferece. Estas características consistem em: criar agentes, fazer com que os agentes executem

tarefas, fazer com que os agentes conversem entre si e publicar e descobrir os “serviços” disponíveis no catálogo de páginas amarelas (BELLIFEMINE; CAIRE, G; GREENWOOD, 2007).

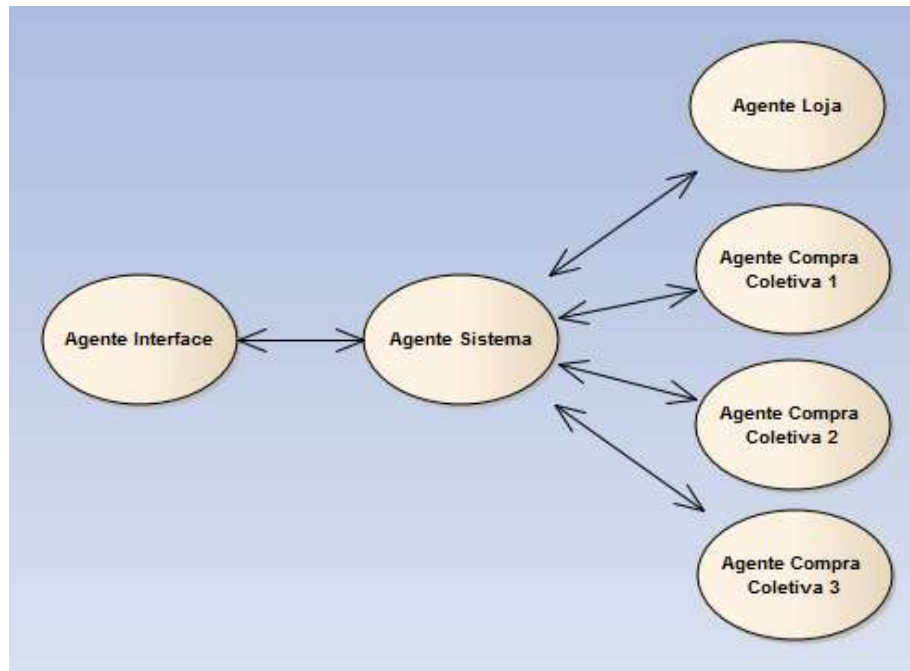


Figura 4.3: Estrutura do sistema multiagentes BuscaCompra

De acordo com o framework JADE, um agente é definido por uma classe que entende a classe `jade.core.Agent`. Para que eles executem tarefas, é necessário criar “Behaviours”. Os comportamentos de um agente representam as tarefas que um agente pode executar e são representados por objetos da classe que estende a classe `jade.core.behaviours.Behaviour`. Foram utilizados apenas dois tipos de comportamentos na aplicação: `OneShotBehaviour`, fornecido pelo framework JADE, onde o comportamento é projetado para executar apenas uma vez; e classes internas de `behaviour` genéricos, onde o desenvolvedor tem a liberdade de definir quando que o comportamento é concluído.

O serviço de páginas amarelas permite que os agentes publiquem serviços para fornecê-los a outros agentes. O agente além de publicar serviços, pode também procurar por serviços de outros agentes. Desta forma, existe a uma troca de serviços entre os agentes. A comunicação feita entre a troca de serviços é considerada a característica mais importante que JADE oferece, pois utiliza todas as especificações FIPA (BELLIFEMINE; CAIRE, G; GREENWOOD, 2007).

4.5.1. Agente Interface

O agente de interface tem como objetivo principal intermediar a aplicação e a estrutura dos agentes. Em termos práticos, este agente recebe a palavra-chave digitada pelo usuário na interface da aplicação e deve retornar uma lista de produtos que contenha aquela palavra-chave.

A Figura 4.4 apresenta um diagrama de classes com foco no agente interface e a sua relação com o framework JADE. Pelo diagrama, podemos observar que o agente interface possui dois tipos de comportamento: informar palavra chave e receber lista de produtos.

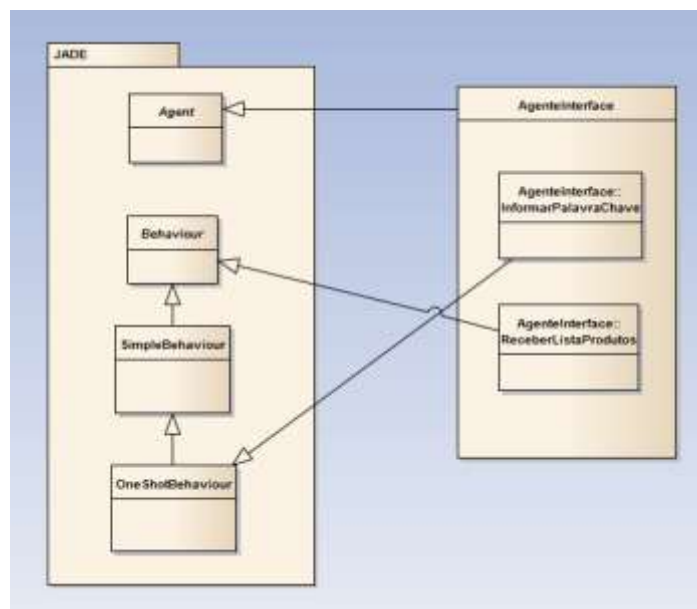


Figura 4.4: Diagrama de classes com foco no Agente Interface

O comportamento “informar palavra chave” envia uma mensagem para o agente sistema, cujo conteúdo é a palavra-chave que foi digitada pelo usuário. Enquanto o comportamento “receber lista produtos” recebe as informações da lista de produtos dos sites de vendas que será enviada pelo agente sistema.

Para que seja possível a comunicação entre o agente interface e o agente sistema em ambos comportamentos, os dois agentes registram seus serviços no serviço de páginas amarelas. Desta forma, o agente interface consegue saber que o agente sistema deseja receber a palavra-chave, podendo enviar a ele esta informação, e o agente sistema consegue saber que o agente interface tem interesse em receber a lista de produtos.

4.5.2. Agente Sistema

O agente sistema é o agente mais importante da estrutura, pois é responsável por emitir a lista de produtos ao agente de interface, criar o agente loja e o agente compra coletiva e gerenciar as informações coletadas pelos mesmos agentes. Sua organização de classes e a relação com o framework JADE é ilustrada na Figura 4.5.

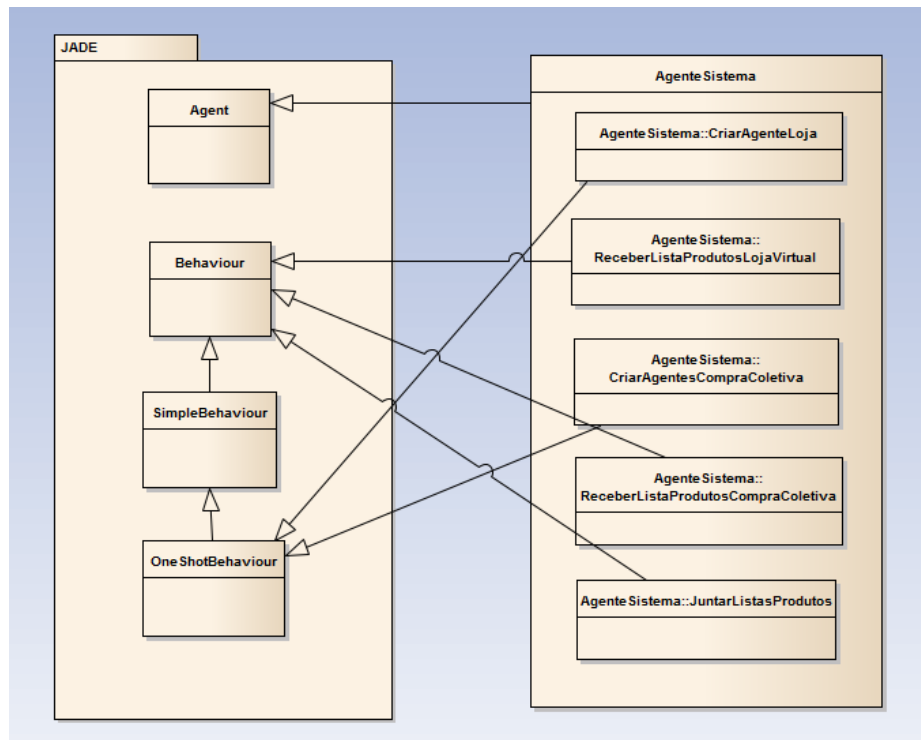


Figura 4.5: Diagrama de classes com foco no Agente Sistema

O agente sistema ao ser criado registra seu serviço no serviço de páginas amarelas. A partir do registro do serviço, é possível a comunicação com o agente interface para enviar a palavra digitada pelo usuário. No momento que o agente sistema recebe esta informação, ele cria toda estrutura necessária para realizar as buscas em sites de venda, através dos comportamentos: criar agente loja e criar agentes compra coletiva.

O agente loja é responsável por toda informação das lojas virtuais, por isso é necessário somente a criação de um único agente através do comportamento criar agente loja. Já o agente compra coletiva é responsável pela informação de apenas um site de compras coletivas, por isso é necessário criar um agente compra coletiva para cada site registrado no sistema.

Cada site de compra coletiva utilizado durante a busca deverá ser previamente registrado no sistema. Este registro utiliza o conceito de wrapper, semelhante à proposta do algoritmo de aprendizagem proposto por (CHOI; HAM; KIM; LEE; YANG, 2000). Este algoritmo de aprendizagem propõe um wrapper que corresponde a uma infraestrutura para extração de informações de uma determinada loja online, responsável por traduzir os códigos HTML em códigos que o agente consiga extrair as informações do produto da respectiva fonte.

Por isso o agente sistema possui o registro de todos os wrappers disponíveis pela aplicação, para poder criar um agente compra coletiva para cada wrapper através do comportamento criar agentes compra coletiva. É importante ressaltar que o único papel do agente sistema é criar um agente compra coletiva para cada wrapper. A utilização desta estrutura é feita pelo próprio agente compra coletiva.

Ao final das buscas, o agente sistema recebe todas informações das lojas virtuais e de todos os sites de compra coletiva, respectivamente, pelos comportamentos: receber lista produtos loja virtual e receber lista produtos compra coletiva. Por fim, ele reúne as informações dos sites de venda em uma única lista e repassa o resultado para o agente interface através do serviço de páginas amarelas.

4.5.3. Agente Loja

O agente loja é criado pelo agente sistema para buscar informações sobre o produto que o usuário deseja buscar em diferentes lojas virtuais. Para realizar seu objetivo, o agente loja conta com dois comportamentos: buscar produtos e retornar lista de produtos. A Figura 4.6 apresenta um diagrama de classes com foco no agente interface e a sua relação com o framework JADE.

Para fazer a busca por produtos, ele interage com o webservice lojas virtuais apenas para consumo do webservice. A função do webservice é disponibilizar uma interface de serviços para que as aplicações se comuniquem com servidores de uma forma mais genérica do que os navegadores da internet (BLAIR; COLOURIS; DOLLIMORE; KINDBERD, 2011).

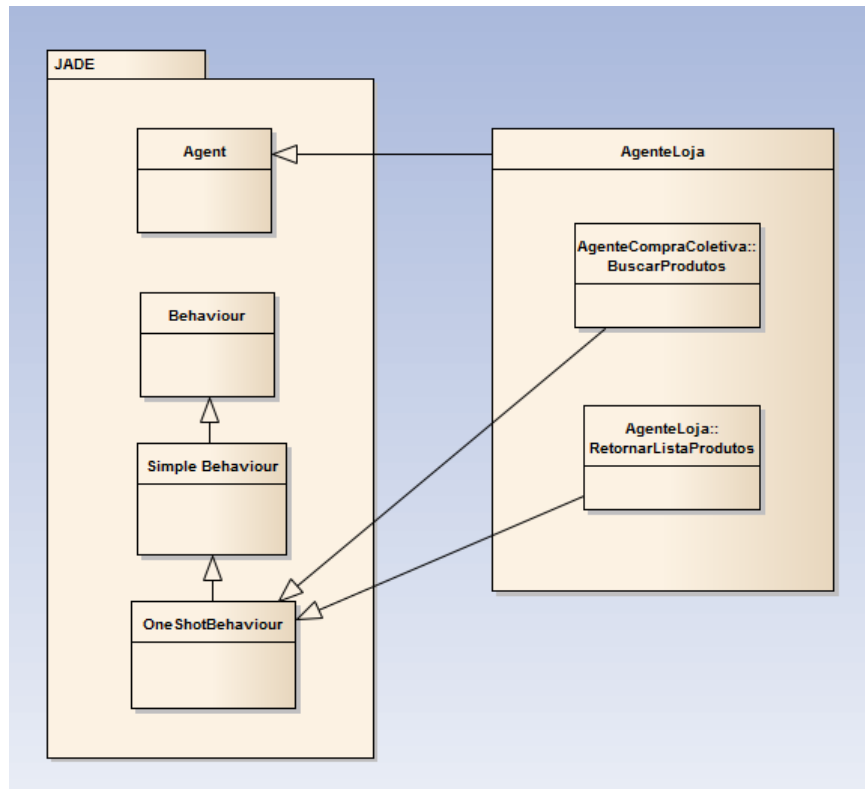


Figura 4.6: Diagrama de classes com foco no Agente Loja

O webservice lojas virtuais é uma abstração do serviço que a empresa Buscapé oferece a desenvolvedores: um webservice com um conjunto de funções e padrões para que sistemas customizados possam acessar sua base de dados de dados e buscar informações de diversas lojas. O agente loja consome o webservice utilizando a busca por palavra chave. A aplicação recebe os dados dos produtos em formato XML e são transformadas em objetos em linguagem java, para então serem repassados ao agente sistema, utilizando o serviço de páginas amarelas.

4.5.4. Agente Compra Coletiva

O agente compra coletiva é responsável pela extração da informação de produtos de um único site de compras coletivas. Ele possui os mesmos comportamentos que o Agente Loja: buscar produtos e retornar lista de produtos. O agente sistema cria inúmeros agentes compra coletiva, um para cada site, onde cada um deles recebe um wrapper correspondente ao site.

Uma das características dos sites de compra coletiva é apresentar todos os produtos de uma cidade em uma só página HTML. Um dos requisitos funcionais do

sistema é utilizar sites de compra coletiva apenas da cidade de Florianópolis. Desta forma, é possível ter acesso a todos os produtos da cidade de Florianópolis em apenas uma página.

Para realizar a pesquisa em sites de compra coletiva, foi criado um crawler focado. Este tipo de crawler é utilizado por inúmeras aplicações que precisam retirar informação em formato HTML de páginas específicas (LUI, 2011). Desta forma, o agente consegue acessar a página do site para manipular seu conteúdo e buscar o produto em questão.

Para a extração das informações em formato HTML é utilizado o wrapper. Como cada site possui uma organização de código HTML diferente, existe a necessidade de criar um wrapper para cada site de compra coletiva utilizado pela aplicação. O principal objetivo é montar uma estrutura que consiga localizar as informações dos produtos, tais como: nome, preço, url da oferta.

Através do crawler focado e do wrapper, o agente compra coletiva consegue ter uma lista de todas as ofertas disponíveis de um respectivo site. Como o objetivo é buscar o produto que foi digitado pelo usuário, o agente compra coletiva percorre todos os produtos disponíveis e verifica se alguma deles contém a palavra-chave. Ao final desta procura, o agente compra coletiva envia o resultado da busca para o agente sistema através do serviço de páginas amarelas.

4.6. Arquitetura

Muitas divisões de um sistema podem ser imaginadas, mas por experiência e convenção, a indústria convencionou a utilização de arquiteturas em camadas e desenvolveu alguns padrões. O princípio das camadas é que elementos de uma camada possuem dependências somente com outros elementos da mesma camada ou dependências dos elementos da camada “abaixo” (FOWLER, 2002).

Para o desenvolvimento do sistema BuscaCompra foi utilizado o princípio das três camadas, proposto por (FOWLER, 2002). O maior objetivo da utilização deste modelo em camadas é favorecer a reutilização do código e o mínimo de acoplamento para que todas as partes da aplicação possam ser reaproveitadas.

Seguindo o desenvolvimento em camadas, a Figura 4.7 mostra as três camadas presentes na aplicação: apresentação, negócio e dados.

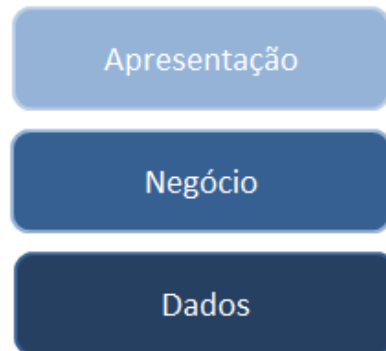


Figura 4.7: Arquitetura do sistema BuscaCompra

A camada de apresentação é responsável por controlar a interação entre o usuário e o software. O principal objetivo desta camada é mostrar as informações ao usuário e interpretar os seus comandos em ações sobre as camadas de negócio e de dados.

A camada de negócio trabalha com a necessidade da aplicação. Ela recebe as informações recebidas pela camada de apresentação, manipula essas informações, e busca os dados da camada de dados.

A última camada do modelo é a camada de dados. Ela proporciona capacidade técnica que suporta as camadas mais altas. Em geral, esta camada corresponde à camada de persistência do domínio. Como nesta aplicação não existe o armazenamento em dispositivo físico, a camada de dados corresponde a toda estrutura dos agentes.

4.7. Desenvolvimento

As camadas de apresentação e negócio foram divididas em três pacotes: presentation, business e entity. A camada de apresentação é representada pelo pacote presentation, onde a classe BuscarProdutoMB é responsável por todas as interações do usuário com a interface gráfica.

A camada de negócio foi dividida em dois pacotes: business e entity. O pacote business possui a classe ProdutoService, responsável por receber informações da camada de apresentação e enviá-las para a camada de dados, além de criar os agentes para realizar a pesquisa. A estrutura dos pacotes é apresentada no diagrama de classes (Figura 4.8). O pacote entity corresponde as entidades do negócio, no caso do sistema BuscaCompra, a única entidade de negócio é o Produto que contém as seguintes informações: nome, preço, URL da oferta e o logotipo da loja.

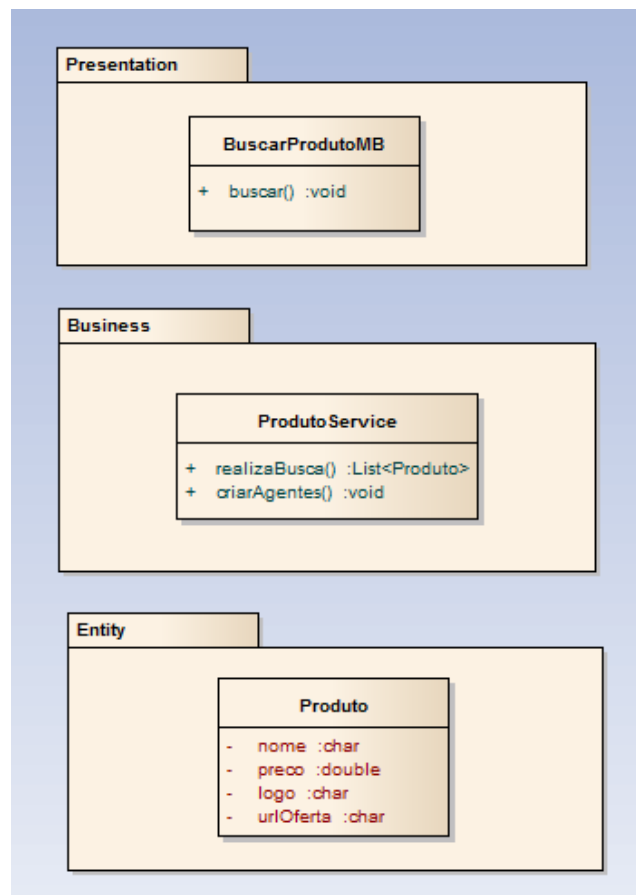


Figura 4.8: Diagrama de classes das camadas de aplicação e negócio

Para ilustrar o funcionamento dessas camadas em tempo de execução, a Figura 4.9 apresenta o diagrama de sequência das duas primeiras camadas.

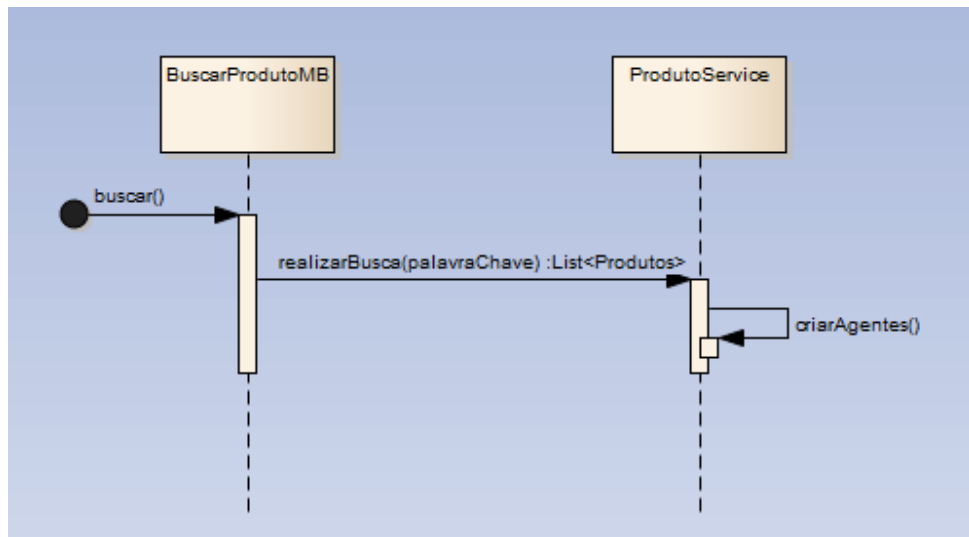


Figura 4.9: Diagrama de seqüência das camadas de aplicação e negócio

A estruturação dos pacotes e das classes correspondentes a camada de dados é apresentada na Figura 4.10. Esta camada é composta por todos os agentes que compõe o sistema BuscaCompra. A estrutura multiagentes, descrita na seção 4.5, é responsável por buscar informações nos sites de venda em tempo de execução.

Além dos agentes, esta camada possui o registro de todos os wrappers. Estes wrappers estão presentes no pacote wrappers, onde cada registro deve estender a classe Wrapper. O objetivo do método getProdutos() é mapear a estrutura HTML de um determinado site para o agente compra coletiva obter a informação de todos os seus produtos.

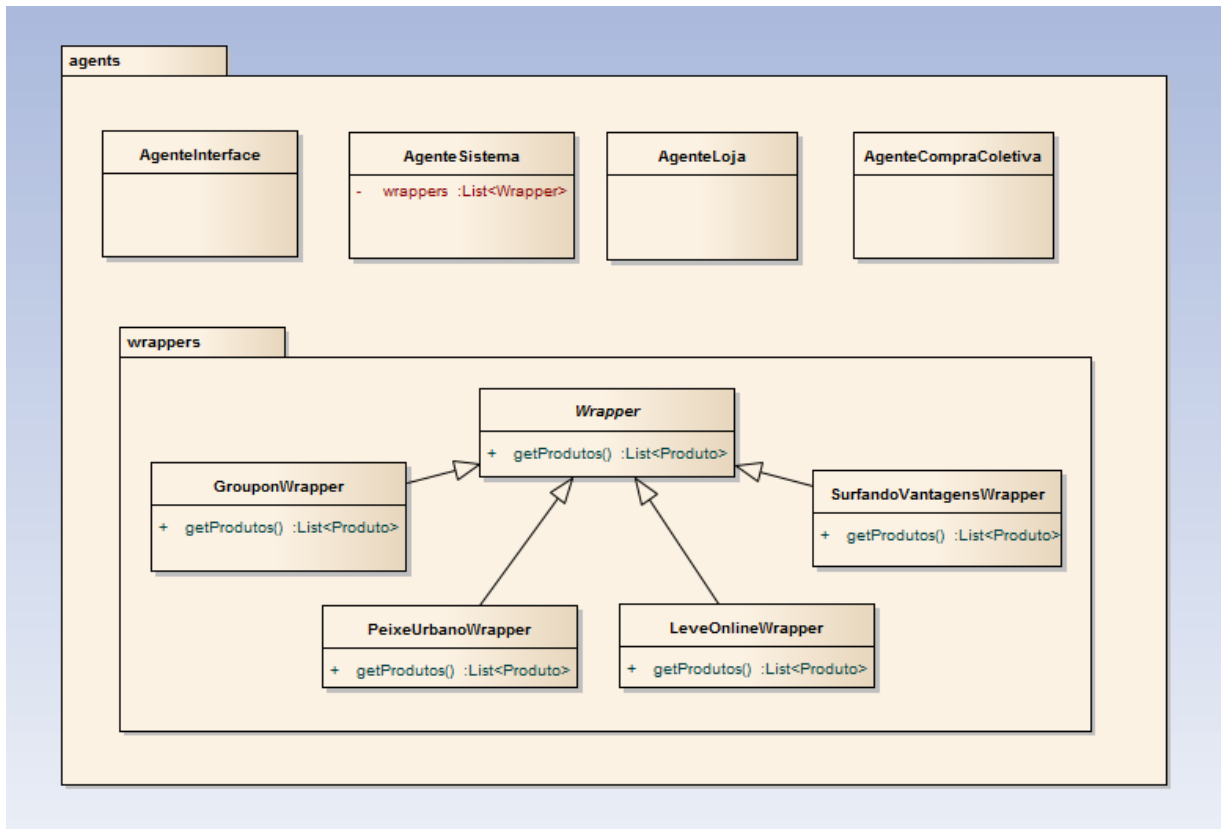


Figura 4.10: Diagrama de classes da camada de dados

4.8. Conclusão

Este capítulo apresentou uma visão geral do sistema proposto por este trabalho, um sistema que reúne informações de produtos em lojas virtuais e sites de compra coletiva. Foram apresentados os requisitos e a modelagem do sistema, bem como descrição dos agentes, decisões de arquitetura e desenvolvimento.

Os testes e as análises de desempenho para a validação da proposta são descritos no próximo capítulo.

5. Avaliação do Sistema

5.1. Introdução

Ao final do desenvolvimento de um sistema, é possível testar as propriedades emergentes, tais como desempenho e confiabilidade. Os testes de desempenho são feitos para assegurar que o sistema possa processar o que foi proposto pelos requisitos. Estes testes de desempenho são extremamente relevantes para sistemas distribuídos, pois estes sistemas coordenam diferentes processos interdependentes, os quais podem se tornar lentos à medida que eles aguardam os dados necessários de outros processos (SOMMERVILLE, 2007).

O objetivo deste capítulo é identificar todas as possíveis interações do usuário com o sistema e avaliar seu tempo de resposta. Através dos testes de desempenho, é possível ter medidas do tempo de resposta das pesquisas e avaliar o desempenho da estrutura de agentes.

5.2. Testes de desempenho

O ambiente de teste criado para a realização dos testes foi uma máquina com o sistema operacional Windows 7 Home Premium, rodando a aplicação localmente utilizando o servidor de aplicação JBOSS, versão 7.0.2, e o browser Google Chrome versão 23.

Para o levantamento dos tipos de testes foi identificado todas as possíveis interações do usuário com o sistema. Como o sistema BuscaCompra possui a interface extremamente simples, apenas com uma caixa de texto e um botão, o usuário não tem muitas possibilidades de interação com o sistema. O foco dos casos de uso está nas possibilidades de texto que o usuário pode digitar, ilustrado na Figura 5.1. Para cada caso de uso ilustrado no diagrama, deverá ser medido o tempo de resposta do sistema.

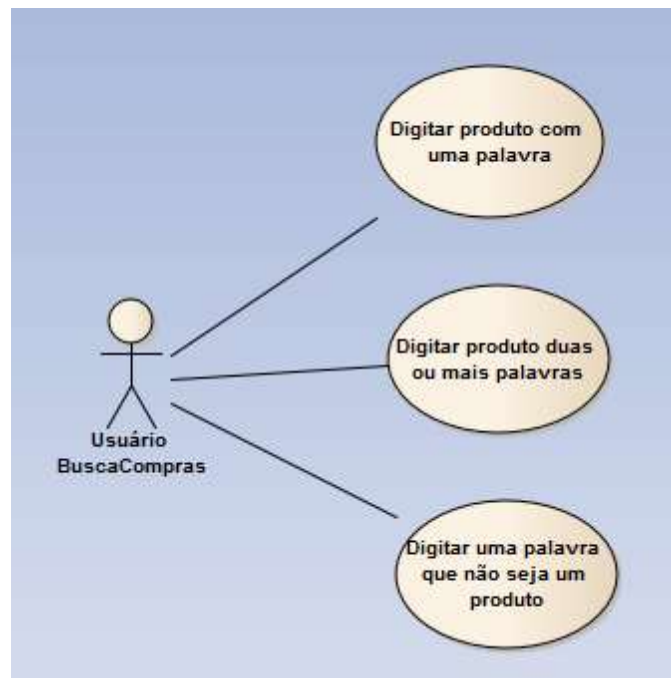


Figura 5.1: Diagrama de casos de uso para teste de desempenho

5.2.1. Execução e Resultados dos Testes

A execução dos testes é feita através de duas buscas para cada caso de uso identificado na Figura 5.1. Para cada teste são registradas as seguintes informações: o caso de uso testado, a palavra chave que foi utilizada para o teste, o tempo de resposta do sistema no momento em que o sistema recebe a informação da palavra-chave e retorna a lista de produtos, a quantidade de produtos de lojas virtuais que foram retornadas pelo resultado da busca e a quantidade de produtos de compras coletivas que foram retornadas pelo resultado da busca.

A Tabela 5.1 mostra os resultados para cada teste. Podemos observar que, independente do tipo de teste, o tempo de resposta do sistema é entre dois e três segundos. O sistema também não apresentou erro de execução durante a realização dos testes. Ao buscar palavras que não são produtos, o sistema mostrou a mensagem de que não foi encontrado nenhum registro.

Tabela 5.1: Resultado da execução dos testes

Caso de uso	Palavra-chave	Tempo de resposta (em segundos)	Qtde produtos lojas virtuais	Qtde produtos compra coletiva
Produto com uma palavra	camera	3	16	3
Produto com uma palavra	bicicleta	2	16	0
Produto com duas ou mais palavras	camera digital	3	16	1
Produto com duas ou mais palavras	Bicicleta aro 26 full suspension	3	16	0
Palavra que não seja um produto	abbbb	2	-	-
Palavra que não seja um produto	ooooopp	2	-	-


As Figuras 5.2 e 5.3 demonstra o sistema ao realizar os testes, a primeira foi utilizada a palavra-chave “camera”, enquanto a segunda utiliza a palavra-chave “bicicleta aro 16 full suspension”.


A Figura 5.2 ilustrou 19 ofertas do produto “camera”, sendo as três primeiras ofertas de sites de compra coletiva e o restante de lojas virtuais. Como a palavra-chave utilizada é mais genérica, as chances de encontrar o produto em uma loja de compra coletiva são maiores, pois o número de produtos disponíveis nestes sites é menor se comparado aos produtos oferecidos por lojas virtuais.


A Figura 5.3 ilustra a realização uma busca mais específica com a palavra “bicicleta aro 16 full suspension”. Por ser uma busca mais específica, é mais difícil encontrar o produto em sites de compra coletiva. O resultado da busca apresentou apenas ofertas de lojas virtuais.

camera

(1 of 2)

 FRETE GRÁTIS! PRONTA ENTREGA! Lindo celular 3Chips, som Stereo 3D, teclado-Qwert, MP3, MP4, Câmera Digital com Flash, Quadriband desbloqueado! (de R\$219,00 por 135,00) de R\$ 219,00 por R\$ 135,00 Você economiza R\$ 84,00 Promoção termina em: 2 vendido(s) 38% off R\$ 84,90 [Compre Aqui!](#)

 Aproveite Tablet 7", 3G, Touch Screen, Android 2.2. Wi-fi, Camera com frete grátis para todo o Brasil! (de R\$599,90 por 290,00) de R\$ 599,90 por R\$ 290,00 Você economiza R\$ 309,90 Promoção termina em: 2 vendido(s) 52% off R\$ 84,90 [Compre Aqui!](#)

 Oferta nacional 28% OFF Câmera Digital Sony Cyber-Shot Face Detection na Eletro Outlet R\$ 379 R\$270 » R\$270 [Compre Aqui!](#)




 Câmera Digital FujiFilm Z700 12.0MP Preta, Zoom Óptico 5x, LCD 3,5" Touch, Detector de Pets e Filmagem de Vídeos em Alta Definição + Cartão SD 2GB FujiFilm 853.20 [Compre Aqui!](#)


Figura 5.2: Teste com a palavra-chave “camera”


bicicleta aro 26 full suspesi

(1 of 2)

 Bicicleta Aro 26 Full Suspension 21 Marchas Freios V - Brake Track & Bikes Boxxer W - 2088403 474.05 [Compre Aqui!](#)

 Bicicleta Aro 26 Full Suspension TB - 100 XS 18 Marchas - Track & Bikes (322248) 399.00 [Compre Aqui!](#)

 Bicicleta Aro 26 Full Suspension TB - 100 XS 18 Marchas - Track & Bikes (322248) 379.05 [Compre Aqui!](#)

 Bicicleta Mormaii 21 Marchas, Aro 26, Full Suspension, Aço Carbono, Rodas em Alumínio - Celebration of One Million Bike Mormaii 398.00 [Compre Aqui!](#)


 Bicicleta Caloi XRT Aro 26 com Full Suspension e 21 Marchas - Preta / Laranja - cod: 27992 548.00 [Compre Aqui!](#)

Figura 5.3: Teste com a palavra-chave “bicicleta aro 16 full suspension”

5.3. Conclusão

O resultado dos testes atingiram as expectativas. O tempo de resposta permaneceu uniforme em todos os possíveis tipos de busca. Este resultado vai ao encontro de uma das vantagens de sistemas multiagentes: a resolução de problemas mais rápida através do paralelismo.

A eficiência de resposta da busca sugere que a utilização de agentes para busca de informações distribuídas pode tranquilamente ser utilizada em aplicações comerciais, onde existe a preocupação de agradar o usuário com o desempenho do sistema.

6. Conclusões e Trabalhos Futuros

6.1. Revisão das Motivações e Objetivos

A programação orientada a agentes é um paradigma de software que abrange os conceitos de teorias da inteligência artificial. A tecnologia de agentes tem sido um assunto extensivo em discussões da comunidade científica por muitos anos, mas apenas recentemente foi visto com importância para aplicações comerciais (BELLIFEMINE; CAIRE, G; GREENWOOD, 2007).

Apesar deste paradigma ser abordado em muitas pesquisas há algum tempo, ainda existem muitas lacunas nas pesquisas que desenvolvem SMA. A maioria dos trabalhos dão ênfase apenas para a parte teórica, deixando de lado os detalhes da implementação do sistema.

Diversas áreas foram estudadas para a elaboração deste trabalho, tais como: agentes inteligentes, sistemas multiagentes, padronização FIPA, plataforma de agentes JADE, metodologia para desenvolvimento de sistemas multiagentes utilizando JADE.

A modelagem do sistema, seu desenvolvimento e a realização de testes fazem parte dos objetivos específicos alcançados por este trabalho. Deste modo, o objetivo geral de construir um SMA para a busca de informações em um ambiente distribuído foi alcançado.

6.2. Principais contribuições

Considerando os objetivos propostos por este trabalho e as atividades desenvolvidas durante este projeto, é possível afirmar que o principal destaque deste trabalho é a união da parte teórica e prática da construção de um sistema multiagente utilizando a plataforma JADE: desde a concepção do sistema, com a utilização de metodologias, até o seu desenvolvimento, com as classes representadas em diagramas.

Além disso, este trabalho contribui através da criação do sistema BuscaCompra: uma ferramenta que permite pesquisar produtos levando em consideração o cenário atual do comércio eletrônico, através da união de mercadorias de lojas virtuais e compras coletivas. Atualmente, não existe nenhuma ferramenta que pesquise em ambos cenários. Este sistema oferece um acesso rápido e personalizado a consumidores exigentes, que realizam pesquisa de mercado antes de realizar a compra.

6.3. Trabalhos Futuros

Considerando o atual estágio deste projeto, algumas possibilidades para a sua continuidade são sugeridas abaixo:

- Utilizar agentes para alimentar uma base de dados;
- Comparar o desempenho da busca distribuída entre sistemas orientados a objetos e sistemas orientados a agentes;
- Propor formas de aperfeiçoar a busca distribuída, sem alterar a estrutura dos agentes proposto por esse projeto;
- Incluir novos tipos de agente ao sistema e avaliar se o desempenho dos agentes propostos por este projeto foi prejudicado com a inclusão dos novos.

REFERÊNCIAS BIBLIOGRÁFICAS

ALHASHEL, E.; BALACHANDRAN, B. M.; MOHAMMEDIAN, M. An Agent-Mediated Collaborative Negotiation in E-Commerce: A Case Study in Travel Industry. Knowledge-Based and Intelligent Information and Engineering Systems. Lecture Notes in Computer Science, p 102-110, 2009.

ALVARES, L. O.; SICHMAN, J. Introdução aos Sistemas Multiagentes. 1997. Jornada de atualização em Informática; Congresso da SBC; Brasília.

ARIDOR, Y.; LANGE, D.B. Agent Design Patterns: Elements of Agents Application Design. Proceedings of the Second International Conference on Autonomous Agents, 1998.

BALAJI, P. G.; SRINIVASAN D. An Introduction to Multi-Agent Systems. Innovations in Multi-Agent Systems and Applications - 1, Studies in Computational Intelligence, p 1-27, 2010.

BELLIFEMINE, F. L.; CAIRE, G; GREENWOOD, D. Developing Multi-Agent Systems with JADE. Wiley, 2007.

BIRUKOV, A.; BLANZIERI, E.; GIORGINI, P. Implicit: an agent-based recommendation system for web search. In: AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems. New York, NY, USA: ACM, 2005. p. 618–624. ISBN 1-59593-093-0.

BLAIR, G.; COULOURIS, G. F.; DOLLIMORE, J; KINDBERD, Jean. Distributed Systems: Concepts and Design. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2011. ISBN 0-13-214301-1.

CAIRE G; BAHRI P.; NIKRAZ, M. *A methodology for the analysis and design of multi-agent systems using JADE*. International Journal of Computer Systems Science and Engineering 21, p. 99-116, 2006.

CHOI, J.; HAM, H.; KIM, H.; LEE, K.; YANG, J. A More Scalable Comparison Shopping Agent. *Proc' Engineering of Intelligent Systems*, pp. 766—772. 2000.

DOORENBOS, R.B.; ETZIONI, O.; WELD, D. A scalable comparison shopping agent for the world-wide web. *Proceedings of the First International Conference on Autonomous Agents*, 39-48, 1997.

ETZIONI, O.; WELD, D. S. Intelligent agents on the internet: Fact, fiction, and forecast. *IEEE Expert: Intelligent Systems and Their Applications*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 10, p. 44–49, 1995. ISSN 0885-9000.

FAUST, F. Todo mundo comprando junto. *Revista Exame*, 01 out. 2010. Disponível em: <http://exame.abril.com.br/revista-exame/edicoes/0975/noticias/todo-mundo-junto-agora-593201>. Acesso em: 1 nov. 2012.

FENG, J.L. Construction of Multi-Agent System for Decision Support of Online Shopping Emerging Research in Artificial Intelligence and Computational Intelligence Communications in Computer and Information Science, p318-324, 2011.

FOWLER, M. *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional, 2002. ISBN: 0321127420.

LIU, B. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer, 2011. ISBN: 3642194591.

NWANA, H.S. Software agents: An overview. *Knowledge Engineering Review*, v. 11, n. 3, p.205–244, 1996.

MENG, B.; ZENG Z. A Multi-Agent Based Intelligent System for Internet Shopping. *Proceedings of International Conference on Services Systems and Services Management*, 2005.

RUSSELL, S.J.; NORVIG, P. *Inteligência artificial*. Rio de Janeiro (RJ): Elsevier, Campus, 2004. ISBN 8535211772.

SOMMERVILLE, I. *Software Engennering*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2007. ISBN: 0-321-31379-8.

VLASSIS, N. A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, v. 1, n. 1, p. 1–71, 2007.

WEISS, M. Patterns for e-Commerce Agent Architectures: Using Agents as Delegates. Pattern Languages of Programming (PLoP-01), 2001.

WOOLDRIDGE, M.; JENNINGS, N.R. Intelligent agent: theory and practice. Knowledge Engineering Review, Cambridge University Press, v. 10, p. 115–152, 1995.

ZUINI, P. 4 Lições do Buscapé para ter um negócio de sucesso. Revista Exame, 23 out. 2011. Disponível em: <<http://exame.abril.com.br/pme/noticias/4-licoes-do-buscapes-para-ter-um-negocio-de-sucesso?page=1>>. Acesso em: 1 nov. 2012.