

UTILIZAÇÃO DE METODOLOGIAS PARA DESENVOLVIMENTO DE AGENTES: UM ESTUDO DE CASO NA MICROECONOMIA

VANESSA M. BERNY, DIANA F. ADAMATTI, DANIELA FERREIRA GOMES,
ANTONIO C. DA ROCHA COSTA

RESUMO

Este artigo apresenta algumas metodologias para engenharia de software orientada para agentes e um estudo de caso aplicado a microeconomia, especificamente em abastecedoras de combustível. Estas metodologias modelam todos os aspectos ligados a agentes, diferentemente de uma metodologia de modelagem orientada a objetos.

Palavras-chave: Engenharia de Software Orientada a Agents,
Microeconomia

Vanessa M. Berny: Mestranda do Programa de Pós-Graduação em Informática - Centro Politécnico - Universidade Católica de Pelotas, vmberny@gmail.com
Dra. Diana F. Adamatti: Centro de Ciências Computacionais - Universidade Federal do Rio Grand, dianadamatti@furg.br
Daniela Ferreira Gomes: Graduanda do Centro de Ciências Jurídicas, Econômicas e Sociais - Universidade Católica de Pelotas, wernhardtdaniela@terra.com.br
Dr. Antonio Carlos da Rocha Costa: do Programa de Pós-Graduação em Informática - Centro Politécnico - Universidade Católica de Pelotas, ac.rocha.costa@gmail.com

Vetor, Rio Grande, v.16, n.2, p. 55-70, 2006.

USING METHODOLOGIES TO AGENT-ORIENTED SOFTWARE ENGINEERING: A CASE STUDY IN MICROECONOMICS

ABSTRACT

This paper presents some methodologies to agent-oriented software engineering and a case study in Microeconomics, specifically in supplies fuel. These methodologies model all aspects of agents approach, differentially of oriented objects approach.

Keywords: Agent-Oriented Software Engineering, Microeconomics

1 INTRODUÇÃO

A escolha da abordagem baseada em Sistemas Multiagentes (SMA) para resolução de problemas tem sido um assunto muito explorado nas últimas décadas. Inúmeros trabalhos têm apresentado conceitualizações, formalizações, protocolos, técnicas e métodos para aplicação deste tipo de abordagem na concepção de softwares. Isso tem acontecido pelo fato da abordagem SMA possuir algumas características que viabilizam a resolução de problemas de outra forma que não a tradicional, adequando-se à problemas complexos (Pressman, 2006) e de natureza descentralizada (Moulin, 1996).

A modelagem de um SMA é bastante complexa. Segundo Bastos (1998), é possível modelar um agente utilizando uma abordagem Orientada a Objetos (OO). No entanto, persiste a problemática da modelagem da sociedade, pois, por não

adequar-se ao modelo conceitual, a abordagem OO não viabiliza a modelagem de todos os aspectos envolvidos em um SMA (Iglesias, 1998; Wooldridge, 1999; Kendall, 1996; Tavater, 1999).

Várias metodologias, arquiteturas e ferramentas já foram desenvolvidas para facilitar o desenvolvimento de sistemas baseados em agentes. As técnicas criadas permitem a construção de aplicações desde as fases de análise e projeto até a implementação final do sistema.

Este artigo apresenta algumas metodologias para modelagem de agentes e um estudo de caso aplicado a microeconomia. O artigo está organizado da seguinte forma: na seção 2 são apresentadas as algumas metodologias para modelagem de SMA; na seção 3 apresenta o estudo de caso, utilizando a metodologia Prometheus. Na seção 4 estão as conclusões deste trabalho.

2 METODOLOGIAS DE DESENVOLVIMENTO DE SISTEMAS MULTIAGENTES

Sob o ponto de vista da Engenharia de Software, a construção de software de alta qualidade de maneira produtiva é viabilizada por um conjunto de métodos, ferramentas e procedimentos. Desta forma, o caminho para a evolução no desenvolvimento de software passa por uma combinação de métodos abrangentes para todas as etapas de desenvolvimento do software, melhores ferramentas para automatizar estes métodos, blocos de construção mais poderosos para a implementação do software, melhores técnicas para a garantia da qualidade do software e uma filosofia de coordenação predominante, controle e administração (Pressman, 2006).

Uma metodologia de desenvolvimento de um SMA deve capturar a flexibilidade, autonomia dos agentes, com variados graus de abstração, auxiliando o projetista nas tomadas de decisão relativas à análise, projeto e implementação.

A distinção entre os requisitos envolvidos no projeto de um SMA e um sistema orientado a objetos (SOO) é evidenciada pelas próprias diferenças existentes entre agentes e objetos. Contudo, o conceito de agente apresenta semelhanças estruturais em relação ao conceito de objeto. O fato é que um agente é uma entidade que possui capacidades comportamentais e conhecimento privado, e que um objeto também possui esta mesma estrutura, demonstra a existência de características comuns entre estes conceitos (Amandi, 1997).

Segundo Shoham (1993), agentes podem ser vistos como objetos ativos com estados mentais, traçando semelhanças entre os dois conceitos, tais como a existência de troca de mensagens entre as entidades para a troca de informações e solicitação de serviços, e os conceitos de herança e agregação.

Existem distinções significativas entre agentes e objetos (Bastos, 1998):

- Agentes são autônomos. Objetos têm autonomia sobre seu estado interno, mas não exibem controle sobre seu comportamento. Assim sendo, objetos têm controle sobre como as coisas são feitas, mas não têm nenhum poder de decidir se determinada solicitação vai ou não ser atendida (Wooldridge, 1999). É preciso observar que nada impede a implementação da característica de autonomia em agentes utilizando-se técnicas OO, porém é necessário notar que a característica de autonomia não é um componente básico do paradigma OO.
- Os agentes têm comportamento autônomo flexível (reativo, pró-ativo, social). Os modelos existentes no paradigma OO não especificam nenhuma maneira de representar estes tipos de comportamento, embora isto possa ser implementado utilizando-se técnicas OO.

Existem algumas metodologias para análise e projeto de SMAs. A seguir, três metodologias serão apresentadas resumidamente.

2.1 Metodologia TROPOS

A metodologia Tropos (Bresciani, 2004) fornece suporte as atividades de análise e de projeto no desenvolvimento do sistema, desde a análise até a implementação do mesmo. Está dividida em cinco fases: fases inicial e final de requisitos, projetos arquitetural e detalhado, e implementação.

Para a análise de requisitos há duas etapas: as fases inicial e final. Na fase inicial são definidos os *stakeholders* do domínio, modelados como atores sociais, com dependências baseadas em objetivos, planos e fornecimento de recursos. Já na fase final, o modelo conceitual é estendido, inclui-se um novo ator que representa o sistema e as dependências com os outros atores do ambiente (Giunchigli, 2002).

As fases de projetos arquitetural e detalhado são relacionadas à especificação do sistema. A fase arquitetural define a arquitetura global do sistema em termos de sub-sistemas (atores), troca de dados e fluxo de controles (dependências). Também

é realizado um mapeamento dos atores do sistema em um conjunto de agentes de software, cada um caracterizado com suas capacidades (Bresciani, 2004). A fase de projeto detalhado especifica as habilidades dos agentes e suas interações. Cada agente é definido, mais especificamente em termos de entrada, saída, controle e outras informações relevantes para o sistema, já nesta fase é escolhida a plataforma de desenvolvimento.

A implementação se baseia na definição do projeto detalhado. Esta metodologia usa uma plataforma para agentes BDI (*Beliefs, Desires and Intentions* - Crenças, Desejos e Intenções), chamada JACK (Jack Intelligent Agents) (JACK, 2005), para a implementação dos SMA. Os agentes em Jack são componentes autônomos que apresentam objetivos a serem alcançados ou eventos a serem tratados. Estes agentes são programados com um conjunto de planos, tornando-os capazes de alcançar seus objetivos.

A modelagem realizada nesta metodologia é bastante confusa e rebuscada, dificultando a fase do processo de desenvolvimento. A fase de projeto detalhado é orientada especificamente à plataforma Jack (Maria, 2005).

2.2 Metodologia GAIA

A metodologia Gaia (Zambonelli, 2003) possui uma linguagem própria para a modelagem de SMA. O processo de desenvolvimento contém duas fases: análise e design. Esta metodologia tem início na fase de análise, visando coletar e organizar a especificação que servirá de base para a fase de design.

A fase de análise tem o objetivo de entender o sistema e decompô-lo em papéis que serão desempenhados na organização, através do modelo de papéis, e definir como os mesmos interagem, através do modelo de interação. Portanto, o modelo de papéis identifica os papéis existentes no sistema e o modelo de interação identifica um conjunto de definições de protocolos, um para cada tipo de interação entre os papéis. Esta fase de análise inclui identificar (Zambonelli, 2003): as metas das organizações presentes no sistema e o comportamento esperado dos mesmos, o ambiente, os papéis iniciais, as interações iniciais e as regras que a organização deve seguir.

Os componentes gerados na fase de análise são utilizados como entrada para a fase de design. Esta fase pode ser decomposta em duas novas fases: a fase da elaboração da arquitetura, que inclui a definição da estrutura organizacional do

sistema em termos de sua topologia e regime de controle e a identificação completa dos papéis e interações; e a fase de detalhamento que compreende a definição do modelo de agentes e a definição do modelo dos serviços que os agentes devem oferecer para desempenhar seus papéis.

Esta metodologia precisa que o SMA seja estático, onde o número de agentes, seus comportamentos, habilidades e as inter-relações não mudam.

2.3 Metodologia PROMETHEUS

A metodologia Prometheus (Padgham, 2004) abrange desde a modelagem até a implementação de um SMA baseado na plataforma para agentes BDI. Esta metodologia é composta por três fases, onde os componentes produzidos são utilizados tanto na geração de esqueleto do código, como também para realização de testes.

A primeira fase corresponde à especificação do sistema e compreende em duas atividades: determinar o ambiente do sistema e determinar os objetivos e funcionalidades do mesmo. O ambiente do sistema é definido em termos de percepções (informações do ambiente) e ações. Além disso, também são definidos dados externos. As funcionalidades do sistema são definidas através da identificação de objetivos, da definição das funcionalidades necessárias para se alcançar esses objetivos e dos cenários de casos de uso (Dam, 2003).

A fase seguinte é o projeto arquitetural que utiliza as saídas da fase anterior para determinar quais agentes existirão no sistema e como os mesmos irão interagir. Esta fase envolve três atividades: definição dos tipos de agentes, definição da estrutura do sistema e definição das interações entre os agentes (Padgham, 2002).

O projeto detalhado é a última fase e é responsável por definir capacidades dos agentes, eventos internos, planos e uma estrutura de dados detalhada de cada tipo de agente identificado na fase anterior.

Atualmente existem duas ferramentas que utilizam o Prometheus. O Jack Development Environment (JDE, 2005), que é um software comercial que inclui uma ferramenta de modelagem para a construção dos diagramas, resultando na geração de código na linguagem de programação Jack (JACK, 2005). Esta ferramenta dá suporte à metodologia Prometheus pelo fato dos conceitos utilizados pelo Jack corresponderem aos componentes gerados na fase de projeto detalhado da metodologia. A outra ferramenta é o Prometheus Design Tool (PDT, 2005). Esta

ferramenta permite que o usuário crie e edite seu projeto utilizando os seguintes conceitos: verificar o projeto para um conjunto de possíveis inconsistências, gerar um conjunto de diagramas de acordo com a metodologia e gerar automaticamente a descrição do projeto, o que inclui descritores para cada entidade, um dicionário para o projeto e um relatório dos diagramas gerados anteriormente, além da geração de um pseudo-código em Jack.

Esta metodologia facilita o aprendizado de sistemas baseados em agentes, pois é bem flexível quanto ao tipo de agentes e possibilita a validação e geração de um esqueleto código, mas poucos trabalhos utilizando esta metodologia foram até o momento publicados.

3 MICROECONOMIA

A Microeconomia, ou Teoria dos Preços, é a parte da teoria econômica que estuda o comportamento das famílias e das empresas e os mercados nos quais operam. Ela preocupa-se mais com a análise parcial, com as unidades (consumidores, firmas, mercados específicos). Já a Macroeconomia estuda os grandes agregados (produto nacional, nível geral de preços, etc.), dentro de um enfoque de análise global (Vasconcellos, 2006).

A empresa não é o foco específico da Microeconomia, mas sim o mercado no qual as empresas e consumidores interagem, analisando os fatores econômicos que determinam tanto o comportamento do consumidor quanto o comportamento do empresa (Vasconcellos, 2006).

Os grandes tópicos abordados na análise microeconômica são os seguintes (Vasconcellos, 2006):

- A Teoria da Demanda ou Teoria da Procura que estuda as diferentes formas que a demanda pode assumir e os fatores que a influenciam.
- A Teoria da Oferta abrange a Teoria da Produção, que estuda o processo de produção numa perspectiva econômica, e a Teoria dos Custos de Produção, que classifica e analisa os custos. A Teoria de Produção envolve apenas relações físicas entre o produto e os fatores de produção, enquanto a Teoria dos Custos também envolve preços dos insumos de produção.
- A Análise das estruturas de mercado aborda a maneira como estão organizados os mercados e como é determinado o preço e a quantidade de equilíbrio nesse mercado.

- A Teoria do Equilíbrio Geral e do Bem-estar estudam a interação de todos os mercados simultaneamente e seu impacto no bem-estar social.
- O estudo das Imperfeições de Mercado aborda as situações nas quais o mercado não promove perfeita alocação de recursos.

3.1 Estudo de Caso: Agentes das Abastecedoras de Combustíveis

Este estudo de caso aborda o que acontece com as abastecedoras de combustíveis. O agente produtor será uma abastecedora de combustível e o agente consumidor será um consumidor de combustíveis que tem necessidades de abastecer seu automóvel periodicamente.

A modelagem conceitual dos agentes foi realizada utilizando a metodologia Prometheus, descrita anteriormente. Os diagramas representam as três fases da metodologia e utilizam os componentes gráficos apresentados na Figura 1. Nesta modelagem, uma Ação é uma atividade a ser realizada pelo agente; um Agente é um indivíduo externo, que interage com o sistema; uma Capacidade é uma função que o agente é capaz de realizar; uma Crença é uma informação que o agente possui e que é atualizada de acordo com as mudanças do ambiente; uma Mensagem é a forma que um agente tem para se comunicar com outros agentes; um Objetivo é uma meta que o agente tentará cumprir; um Plano é um conjunto de ações para realizar o objetivo; e um Protocolo é a formalização de como as mensagens entre agentes são enviadas/recebidas.



Figura1: Componentes gráficos utilizados na Metodologia Prometheus

O diagrama da Figura 2 faz parte da hierarquia de concepção de diagramas da fase de especificação do sistema, que tem o intuito de apresentar os agentes produtor e consumidor, apresentando suas interações (protocolos de mensagens), suas capacidades e ações possíveis. Os principais objetivos de cada um dos agentes são:

- Consumidor: tem o principal objetivo de abastecer. Para alcançar este objetivo este agente deve realizar as ações pagar e de aceitar ou rejeitar propostas de valores de venda.
- Produtor: tem o principal objetivo de prestar_servico ao consumidor. Para alcançar este objetivo este agente realiza a ação de fornecer_produto, e de aceitar e rejeitar propostas. Também tem objetivo de localizar_pto-equilibrio, que é a possibilidade de encontrar o equilíbrio do mercado, onde a oferta do produtor é igual a demanda do consumidor.

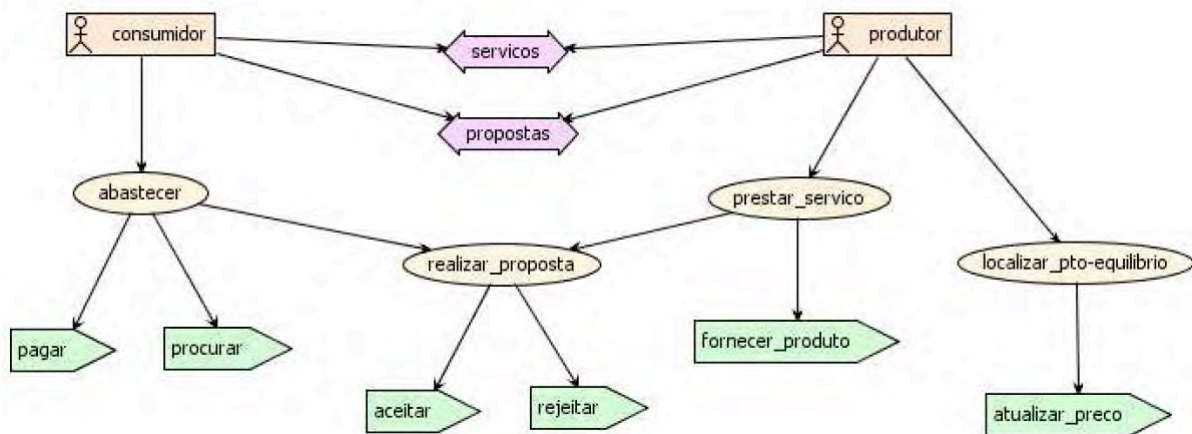


Figura 2: Fase de Análise - Visão Geral do Sistema

Na Figura 3 é apresentado um dos diagramas da fase de projeto arquitetural da modelagem, que apresenta uma visão geral de todo o sistema, envolvendo os agentes produtor e consumidor. Para esta fase, são apresentadas as percepções dos agentes (percepções ou crenças são informações que o agente possui dele próprio ou do ambiente em que ele está inserido), ações (realizadas para alcançar objetivos) e os protocolos de interações (trocas de mensagens que ocorrem entre os agentes).

Para o agente consumidor tem-se as seguintes percepções:

- Restricao_orcamentaria: principal fator decisório para as escolhas de comprar um produto;
- Qualidade_servicos: fator relevante quando a restrição orçamentária for irrelevante ou pouco relevante e o consumidor ter uma demanda de serviços com qualidade;

- Distancia_localizacao: fator em relação a distância, o estabelecimento próximo a residência, local de trabalho do consumidor, ou por onde ele passe com frequência;
- Lealdade_com_produto: fator importante que todo produtor gostaria de ter, um estabelecimento como referência, um consumidor assíduo aos seus produtos e serviços;
- Qualidade_produto: fator relevante quando a restrição orçamentária for irrelevante ou pouco relevante e o consumidor ter uma demanda de produtos com qualidade;
- Dinheiro: armazena o fator moeda que o consumidor possui e pode gastar para comprar produtos;
- Formas_pagto: fator relevante para decisão do consumidor de adquirir um produtor de forma que possa pagar com suas condições;
- Preço_produto: fator que para alguns consumidores pode ser decisivo para a realização da compra ou serviço;
- Apresentacao_estabelecimento: fator importante para o consumidor que presa contar com serviços/produtos de estabelecimentos limpos e bem decorados;
- Quantidade_escolhas_possiveis: armazena a quantidade de estabelecimentos que podem ser escolhidos no momento de necessidade do produto;
- Grafico_consumidor: cada consumidor possui um gráfico individual baseado na microeconomia, que será consultado na análise das propostas;
- Informacoes_propostas_consumidor: armazena informações trocadas das propostas do consumidor.

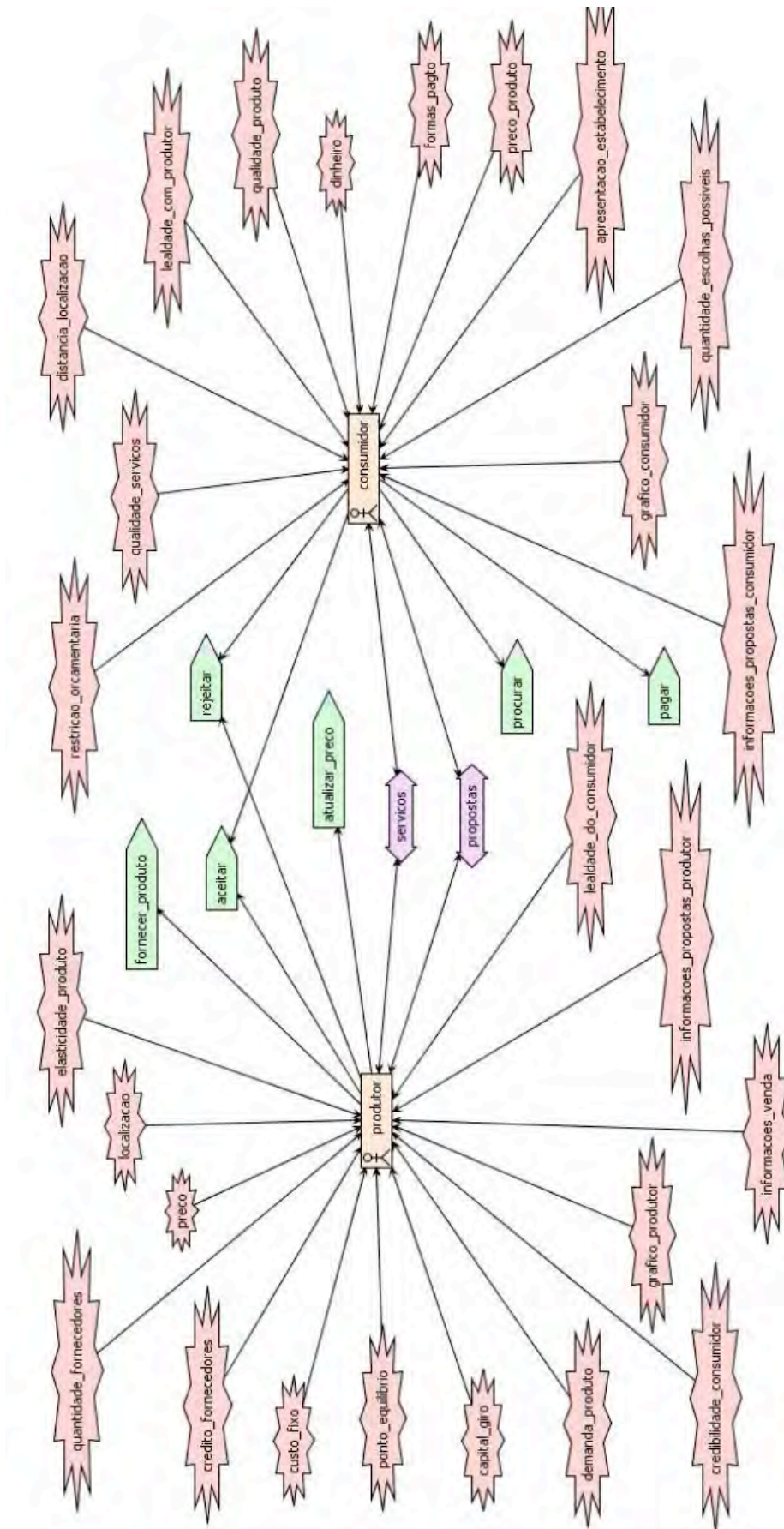


Figura 3: Projeto Arquitetural - Visão Geral do Sistema

Para o agente consumidor tem-se as seguintes ações:

- Aceitar: ação de aceitar uma proposta vinda do produtor para depois realizar o pagamento do produto;
- Rejeitar: ação de rejeitar uma proposta vinda do produtor;
- Procurar: ação de procurar por um produtor que atenda suas necessidades;
- Pagar: ação de realizar o pagamento em moeda dos serviços prestados pelo produtor.

Já o agente produtor tem as seguintes percepções:

- Elasticidade_produto: os produtos podem ser classificados como elásticos (produtos que quando seu preço aumenta a demanda diminui, são também denominados de produtos supérfluos) ou inelásticos (produtos que não têm sua demanda diminuída na mesma proporção do aumento de preço, também são chamados de produtos necessários, básicos ou de subsistência);
- Localização: a localização do estabelecimento pode influenciar sobre os preços cobrados pelo produto;
- Preço: armazena o valor atual do produto comercializado;
- Quantidade_fornecedores: fundamental para os comerciantes que trabalham com produtos elásticos, pois ele irá definir o poder de barganha do produtor;
- Credito_fornecedores: influencia no preço final do produto, porque se um estabelecimento tiver uma boa fama na praça, quer dizer que este é considerado um bom pagador;
- Custo_fixo: são praticamente constantes e deverão ser considerados no cálculo do valor dos produtos ofertados; Ponto_equilíbrio: é o ponto onde os custos totais se igualam à receita da empresa, pois só depois de ultrapassar o ponto de equilíbrio o produtor passará a ter lucro;
- Capital_giro: é importante para produtores que não possuem crédito com seus fornecedores;
- Demanda_produto: média de quantidade de produto que os consumidores geram em um determinado de tempo;
- Credibilidade_consumidor: fator que todo e qualquer produtor almeja, é diretamente ligado ao tempo em que este atua no mercado e a forma de atuação;
- Grafico_produto: cada produtor possui um gráfico individual baseado na microeconomia, que será consultado na análise das propostas;

- Informacoes_venda: armazena informações sobre cada venda realizada, contento a quantidade das vendas e o valor em moeda do que foi vendido;
- Informacoes_propostas_produutor: armazena informações trocadas das propostas do produtor;
- Lealdade_do_consumidor: é a meta de todo produtor, porém para alcançá-la este deve conquistar o consumidor.

As ações do agente produtor são:

- Fornecer_produto: ação de fornecimento do produto para o consumidor;
- Aceitar: ação de aceitar uma proposta vinda consumidor para depois realizar o fornecimento do produto;
- Rejeitar: ação de rejeitar uma proposta vinda do consumidor;
- Atualizar_preco: ação de baixar e aumentar o preço do produto, conforme a Lei da Oferta e da Demanda, quando for necessário.

Além dos diagramas apresentados nas Figuras 1 e 2, foram modelados os diagramas detalhados dos agentes consumidor e produtor, identificando as crenças com as ações e os protocolos de comunicação.

4 CONCLUSÕES

Este artigo apresentou metodologias para modelagem a SMA, baseadas na Engenharia de Software, bem como um estudo de caso da metodologia Prometheus aplicado a Microeconomia.

Existe uma gama de padrões e conceitos voltados para as técnicas de estruturação da modelagem conceitual de um SMA. As metodologias existentes estão suprindo as necessidades de se realizar uma modelagem bem estruturada, uma vez que as mesmas estão captando a flexibilidade, autonomia dos agentes, com variados graus de abstração, auxiliando o desenvolvedor nas tomadas de decisão em relação à análise, projeto e implementação do SMA.

Um ponto importante a ressaltar é que o estudo de caso das abastecedoras de combustíveis modelado a partir de uma metodologia de modelagem para SMA foi possível visto que agentes de software possuem objetivos a serem alcançados e interações que ocorrem entre estes. Isso não seria possível uma modelagem baseada em orientação a objetos, uma vez que objetivos e interações não podem ser representados como métodos, caso o agente seja considerado uma classe.

REFERÊNCIAS BIBLIOGRÁFICAS

A. A. AMANDI. Programação de Agentes Orientada a Objetos. Tese de doutorado, Universidade Federal do Rio Grande do Sul, 1997.

R. M. BASTOS. O Planejamento de Alocação de Recursos Baseado em Sistema Multi-Agentes. Tese de doutorado, Universidade Federal do Rio Grande do Sul, 1998.

P. BRESCIANI, A. PERINI, P. GIORGINI, F. GIUNGHIGLIA, and J. MYLOPOULOS. Tropos: An agent-oriented software development methodology. *Journal of Autonomous Agents and Multi-Agent Systems*, 08(01):203–236, 2004.

K. H. DAM and M. WINIKOFF. Comparing agentoriented methodologies. In *International BiConference Workshop on Agent-Oriented Information Systems*, pages 78–93, 2003.

F. GIUNGHIGLI, J. MYLOPOULOS, and A. PERINI. The tropos software development methodology: Processes, models and diagrams. In *International Workshop on Agent-Oriented Software Engineering*, pages 162–173, 2002.

C. A. IGLESIAS, M. GARIJO, J. C. GONZALEZ, and J. R. VELASCO. Analysis and design of multiagent systems using mas-commonkads. In *Lecture Notes in Computer Science*, v.1365, pages 313–327, 1998.

JACK. Jack intelligent agents, 2005. Disponível em: <http://www.agentsoftware.com/products/jack/>, acesso em: setembro de 2005.

JDE. Jack development environment, 2005. Disponível em: <http://www.agentsoftware.com.au/shared/products/whatsnew50.html>, acesso em: setembro de 2005.

E. A. KENDALL, M. T. MALKOUN, and C. JIANG. A methodology for developing agent-based systems. In Lecture Notes in Computer Science, v.1087, pages 85–99, 1996.

B. A. D. MARIA. Usando a abordagem MDA no desenvolvimento de sistemas multi-agentes. Dissertação de mestrado, Pontifícia Universidade Católica do Rio de Janeiro, 2005.

B. MOULIN and B. CHAIB-DRAA. An Overview of Distributed Artificial Intelligence. John Wiley and Sons, London, 1996.

L. PADGHAM and M. WINIKOFF. Prometheus: A pragmatic methodology for engineering intelligent agents. In Workshop on Agent-Oriented Methodologies, pages 97–108, 2002.

L. PADGHAM and M. WINIKOFF. Developing Intelligent Agent Systems: A Practical Guide. RMIT University, 240p, Melbourne, 2004.

PDT. Prometheus design tool, 2005. Disponível em: <http://www.cs.rmit.edu.au/agents/pdt/>, acesso em: setembro de 2005.

R. S. PRESSMAN. Engenharia de Software. McGraw-Hill, 6a. edition, 2006.

Y. SHOHAM. Agent-oriented programming. In Elsevier Science Publishers, Artificial Intelligence, v.60, pages 51–92, 1993. K. Taveter. Business Rules' Approach to the Modelling, Design, and Implementation of Agent-Oriented Information Systems. VTT Information Technology, p.317-335, Finland - FIN, 1999.

M. A. S. VASCONCELLOS. Economia: Micro e Macro. Editora Atlas, São Paulo, 4a. edition, 2006. 441 pp.

M. WOOLDRIGE. Intelligent agents. In G. Weiss, editor, Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence, chapter 1, pages 27–78. The MIT Press, 1999.

F. ZAMBONELLI, N. R. JENNINGS, and M. WOOLDRIDGE. Developing multiagent systems: The gaia methodology. CM Transactions on Software Engineering and Methodology, pages 317–370, 2003.