Proposta de uma arquitetura para construção de Objetos Inteligentes de Aprendizagem baseados em agentes BDI

Natanael R. Bavaresco[1], Ricardo A. Silveira[1]

[1]Departamento Informática e Estatística – Universidade Federal de Santa Catarina
88040-900 – Florianópolis, SC – Brasil
{natanael,silveira}@inf.ufsc.br

Abstract. *The Learning Objects are a great initiative to increase the reusability of educational material. Despite this, the e-learning systems are not prepared to attend the individual learner's characteristics or provide custom learning experiences. This paper proposes the design of a multi-agent system based on Belief-Desire Intention (BDI) paradigm to create Intelligent Learning Objects (ILOs). The system is supported by techniques of Software Engineering oriented to the agents for the development of agent society.*

Keywords: BDI Agents; Intelligent Learning Objects, Software Engineering

Resumo. *Os Objetos de Aprendizagem são uma grande iniciativa para aumentar a reusabilidade de material educacional. Contudo os sistemas de e-learning baseados em objetos de aprendizagem não são ainda preparados suficientemente para atender as características individuais dos estudantes ou possibilitando experiências de aprendizagem individualizadas. Este artigo propõe um modelo de agentes baseados no paradigma Belief-Desire-Intention (BDI) suportado por técnicas de Engenharia de Software Orientada a Agentes para o desenvolvimento de Objetos Inteligentes de aprendizagem.*

Palavras chaves: Agentes BDI; Objetos Inteligentes de Aprendizagem; Engenharia de Software

1. Introdução

De acordo com [Silveira et al. 2005], um Objeto Inteligente de Aprendizagem é uma entidade que consiste de um agente capaz de desempenhar o papel de um objeto de aprendizagem. [Silva et al. 2008] considera que objetos inteligentes podem comunicar-se com um repositório de objetos e com um sistema gerenciador de aprendizagem, todos eles implementados na forma de agentes, os quais compreendem, em conjunto, um ambiente de aprendizagem.

O presente artigo apresenta a continuidade de pesquisas que vêm sendo desenvolvidas relacionadas a objetos inteligentes de aprendizagem, onde: [Silveira et al. 2005] definiu o conceito e diálogos necessários para uma arquitetura de agentes que permita a interoperabilidade entre objetos de aprendizagem; [Silva et al. 2008] apresentam um framework de sistema multi-agente onde objetos inteligentes de aprendizagem construídos sob a especificação SCORM [ADL 2004] possam interagir.

A partir destes trabalhos anteriores, propõe-se agora uma nova abordagem através da utilização de uma arquitetura de agentes BDI, imprimindo maior poder representacional aos agentes e incrementando sua capacidade de decisão.

2. Agentes BDI

O modelo BDI (*belief-desire-intention*) [Rao e Georgeff 1995] possibilita representar o conhecimento e alguns aspectos de comportamento de agentes de software quando este se assemelha ao raciocínio humano. O raciocínio descrito através da lógica formal é a base para ações que selecionam um conjunto de desejos em função da crença dos agentes e como esses desejos concretos produzidos, como resultado do passo anterior, podem ser atingidos através do uso dos meios disponíveis ao agente [Wooldridge, 2000]. Os estados mentais no modelo BDI podem ser definidos como: Crenças (*Beliefs*): representam as características do ambiente, as quais são atualizadas apropriadamente após a percepção de cada ação; Desejos (*Desires*): contêm informação sobre os objetivos a serem atingidos, bem como as prioridades e os custos associados com os vários objetiv*os;* Intenções *(Intentions*): representam o atual plano de ação escolhido. Capturam o componente deliberativo do sistema.

3. Desenvolvimento do Modelo

A sociedade de agentes foi modelada utilizando uma metodologia própria para sistemas multi-agentes, baseada em conceitos da engenharia de software. A metodologia escolhida foi à metodologia Prometheus, a qual, pela sua simplicidade, facilita o entendimento e possibilita focar o real objetivo do sistema que se deseja construir. Por prover uma interligação entre os modelos, é possível estabelecer vínculos em todas as etapas, garantindo que todos os objetivos identificados sejam atendidos pelos agentes e diálogos entre eles.

3.1 Metodologia Prometheus

Prometheus [Padgham e Winikoff 2002] é uma metodologia para desenvolvimento de sistemas multiagentes que abrange desde a modelagem até a implementação. Esta metodologia é composta por três fases, onde os artefatos produzidos são utilizados tanto na geração do esqueleto do código, como também para depuração e teste.

A primeira fase, correspondente à especificação do sistema, compreende duas atividades: determinar o ambiente do sistema e determinar os objetivos e funcionalidades do sistema. O ambiente do sistema é definido em termos de percepções (informações provenientes do ambiente) e ações. Além disso, 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.

A fase seguinte (projeto arquitetural) 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.

A última fase desta metodologia (projeto detalhado) 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.

A metodologia foi escolhida por ser de fácil entendimento sendo por isso utilizada tanto no meio acadêmico quanto industrial. A ferramenta de desenvolvimento PDT (Prometheus Design Tool) é de fácil utilização e permite um claro entendimento dos diagramas.

3.2 *Framework* Jadex

Jadex [Pokahr, Braubach e Lamersdorf 2003] é um mecanismo de raciocínio orientado a agentes, no qual agentes racionais são escritos em XML e na linguagem de programação Java. Ao invés de implementar uma nova linguagem, os agentes Jadex podem ser programados nos ambientes de desenvolvimento integrado orientados a objetos.

Outro conceito importante é a independência do Jadex. Como ele é fracamente acoplado com o *middleware* sobre o qual ele é executado, Jadex pode ser usado com diferentes plataformas de agentes. Atualmente, dois adaptadores estão disponíveis. O primeiro é a plataforma JADE [Bellifemine, Claire e Greenwood 2007], a qual é bem conhecida e de código livre. O segundo é o adaptador Jadex Stand alone, o qual é um ambiente pequeno, mas rápido.

Jadex segue o modelo BDI, utilizando crenças, objetivos e planos com os objetos de primeira classe, que podem ser criados e manipulados dentro do agente. No Jadex, agentes têm crenças, que são armazenadas em uma base de crenças. Objetivos representam motivações concretas, como por exemplo, estados a serem atingidos, e que influenciam no comportamento do agente. Para atingir seus objetivos, o agente executa seus planos, os quais são procedimentos codificados em Java.

4. Modelagem

A metodologia Prometheus provê diversos diagramas em suas fases que permitem um detalhamento do sistema que pretende se construir. O Diagrama de Objetivos permite definir os principais objetivos e sub-objetivos que devem ser alcançados pelo sistema.

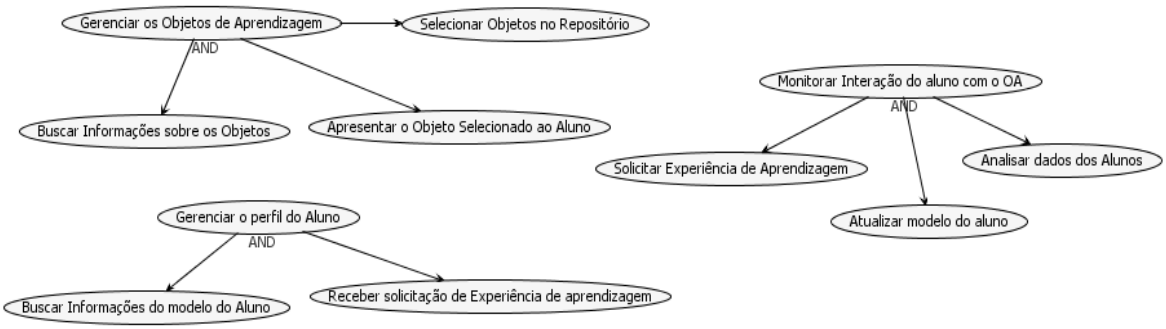

Figura 1. Diagrama de Objetivos do Sistema

O diagrama central é o Diagrama de Visão Geral aonde é possível observar os agentes, seus papéis, percepções, mensagens e ações. Este diagrama permite ter uma idéia geral do funcionamento do sistema facilitando o entendimento e adição de novas funcionalidades.

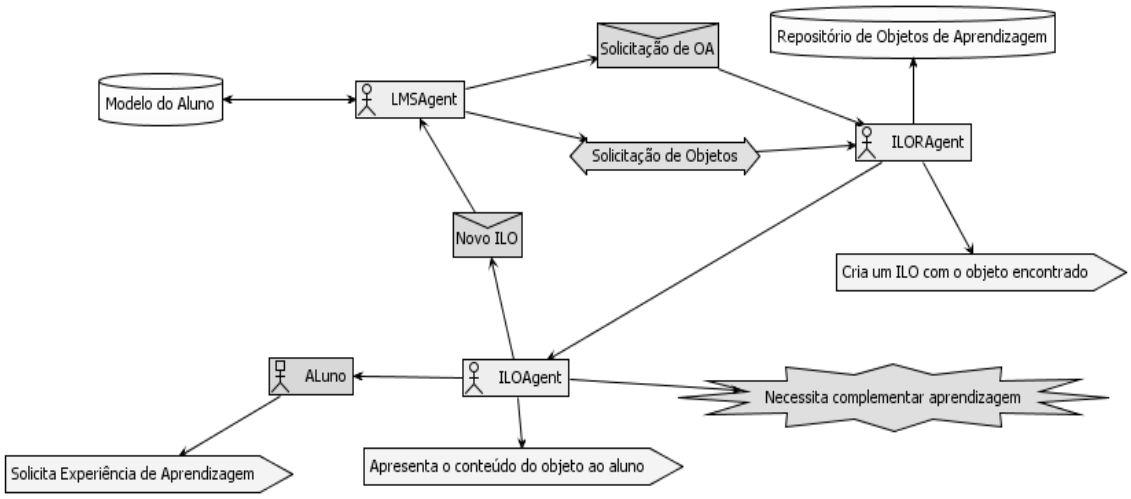

Figura 2. Visão Geral do Sistema

5. Arquitetura Proposta

A Arquitetura proposta neste trabalho consiste de um servidor web, um repositório de objetos de aprendizagem e um servidor que hospedará os agentes. Desta maneira, o aluno poderá acessar o sistema e requisitar objetos para uma experiência de aprendizagem. O Servidor web irá hospedar as páginas web e servlets da aplicação. Todo o raciocínio será realizado entre os agentes através do processo de deliberação. O servidor de repositório irá armazenar os objetos de aprendizagem aonde os agentes serão capazes de buscar e recuperar os objetos adequados às necessidades dos alunos. A figura 3 mostra uma visão geral de todo o sistema.

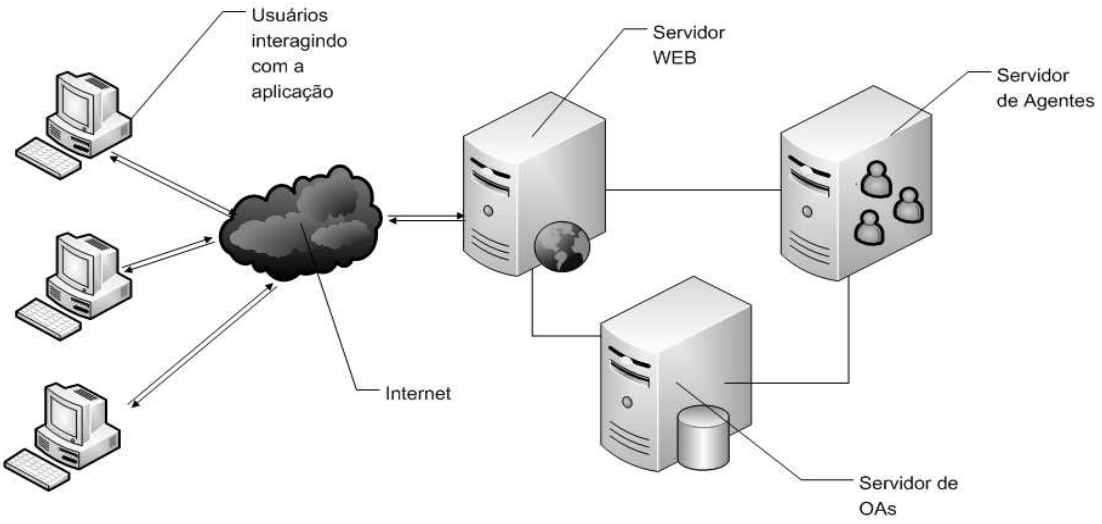

Figura 3. Visão Geral da Arquitetura

5.1. Desenvolvimento

O SMA foi desenvolvido através do framework Jadex na versão 0.96, o qual provê os serviços básicos para criação e utilização dos agentes, oferecendo uma infraestrutura de comunicação e gerenciamento. Desse modo é possível criar agentes BDI definindo suas crenças, objetivos e planos em arquivos XML chamados ADF (Agent Definition File). Após definir planos, que serão executados para atingir os objetivos, é necessário codificar suas ações em classes Java.

O framework Jadex ainda possibilita a integração entre agente e sistemas web através de um add-on chamado Webbridge. Ele permite a comunicação entre agentes e Servlets. A comunicação inicia com um servlet chamado DelegateServlet que recebe uma requisição http e traduz para os agentes através de mensagens ACL. Após isso, o CoordinatorAgent, disponibilizado pelo Webbridge, recebe as mensagens e cria um novo agente que inicia a execução dos planos.

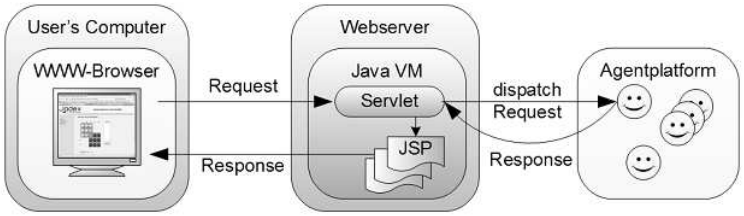

Figura 4. Jadex Webbridge (Webbridge Guide)

Quando o aluno começa a interagir com o sistema, ele tem suas informações consultadas pelo LMSAgent (abstração do Sistema Gerenciador de Aprendizagem) que atualiza sua base de crenças e começa a gerar objetivos. Após definido o objetivo ele começa a executar os planos do objetivo escolhido como solicitar ao agente ILORAgent (Abstração do Repositório) objetos de aprendizagem. Após selecionar o objeto de aprendizagem adequado, um agente ILOAgent (abstração de Objeto de Aprendizagem) será criado e terá a responsabilidade de interagir com o estudante e monitorar essa interação recuperando dados do modelo de dados SCORM através da sua API. Durante esta interação entre o aluno e o objeto representado pelo ILOAgent, este pode estabelecer um novo objetivo como complementar a aprendizagem, devido à demora do aluno em interagir com o objeto por exemplo. Neste caso o ILOAgent começa a trocar mensagens com o ILORAgent e LMSAgent solicitando um novo objeto. O LMSAgent atualiza as informações sobre o aluno na sua base de crenças e inicia novamente o ciclo solicitando ao ILORAgent um novo objeto de aprendizagem.

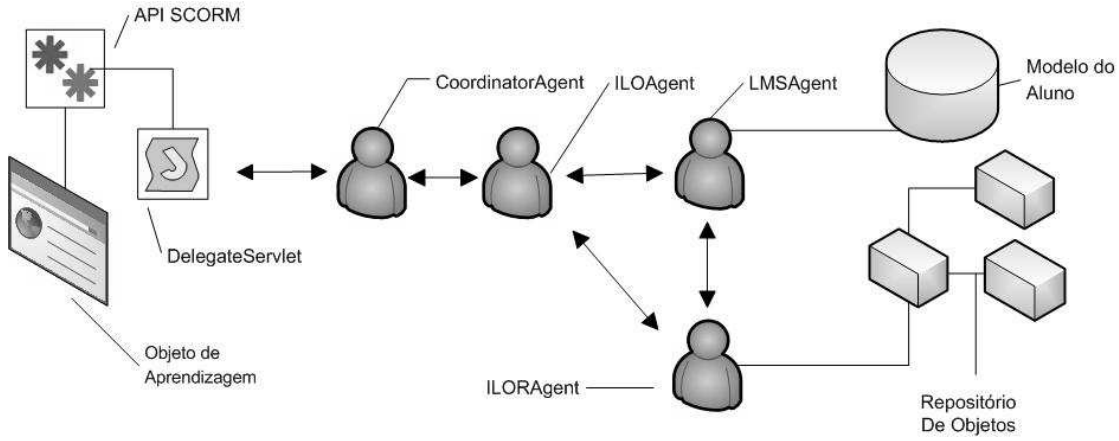

Figura 5. Proposta da arquitetura

6. Conclusões

Este artigo mostra uma arquitetura baseada em trabalhos anteriores como [Silva 2007] e [Gomes 2005] propondo uma nova abordagem de Objetos Inteligentes de Aprendizagem utilizando agentes BDI. Esses agentes raciocinam através de estados mentais e podem atualizar suas crenças em tempo de execução. Desse modo o sistema de aprendizagem pode se tornar mais adaptativo em relação às abordagens anteriores, pois os agentes BDI podem mudar suas crenças sobre o ambiente e definir novos objetivos em tempo de execução possibilitando novas experiências de aprendizagem. Além disso, o framework Jadex possibilita a fácil mudança do comportamento dos agentes através do arquivo de definição, sendo que novos planos podem ser adicionados aumentando as suas capacidades.

O uso de metodologias de software orientadas a agentes auxilia a estabelecer os objetivos, papéis, capacidades e interações entre os agentes. A metodologia utilizada

além de fornecer diagramas de fácil entendimento, fornece também suporte completo à criação de sistemas de agentes BDI.

O framework Jadex possibilita a criação e monitoramento de sistemas multi-agentes complexos através de seus recursos e ferramentas. Através dele é possível criar facilmente agentes com comportamentos complexos e que podem se comunicar com uma interface web, o que facilita o acesso do usuário ao sistema e abstrai os detalhes do funcionamento.

Para futuros trabalhos a aplicação será finalizada, bem como a arquitetura interna de cada agente será aprofundada e testes adicionais com mais objetos de aprendizagem no padrão SCORM, principalmente em ambientes diferentes distribuídos em uma rede.

Referências

Advanced Distributed Learning (ADL). Sharable Content Object Reference Model (SCORM®) 2004 Overview. www.adlnet.org, Acesso em Agosto, 2009.

Gomes, E. R. "Objetos Inteligentes de Aprendizagem: uma abordagem baseada em agentes para objetos de aprendizagem". Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre (2005)

Padgham, L. and Winikoff, M. Prometheus: A pragmatic methodology for engineering intelligent agents. In Proceedings of the OOPSLA 2002 Workshop on Agent-Oriented Methodologies, pages 97-108, Seattle (2002).

Pokahr, A.; Braubach L., Lamersdorf, W. Jadex: A BDI Reasoning Engine, Chapter of Multi-Agent Programming, Kluwer Book, Editors: R. Bordini, M. Dastani, J. Dix and A. Seghrouchni. (2003).

Rao, A. and Georgeff M. "BDI Agents: from theory to practice". In V. Lesser, Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95), pgs 312–319. The MIT Press: Cambridge, MA, USA (1995).

Silveira, R. A.; Gomes, E. R.; Vicari, R. "Improving Interoperability Among Learning Objects Using FIPA Agent Communication Framework". In: IFIP World Computer Conference - WCC, 2006, Santiago. Professional Practice in Artificial Intelligence. Berlin: Springer, (2006).

Silva, J.M.C. "Desenvolvimento de um framework para Objetos inteligentes de aprendizagem Aderente a um modelo de referência para Construção de conteúdos de aprendizagem". Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis (2007)

Silva, J. M. C.; Bavaresco, N.; Silveira, R. A. "Projeto e Desenvolvimento de um Sistema Multiagentes para Objetos Inteligentes de Aprendizagem Baseado no Padrão SCORM". Revista Brasileira de Informática na Educação. (2008).

Wooldridge, M. J. Reasoning about Rational Agents, MIT Press (2000).