Integrando agentes AgentSpeak(L) em ambientes pervasivos educacionais

Alexandre de O. Zamberlan[1], Reiner F. Perozzo[2], Guilherme C. Kurtz [2] Giovani R. Librelotto[3], Solange B. Fagan[1]

[1]Programa de Pós-Graduação em Nanociências
Centro Universitário Franciscano

[2]Curso de Ciência da Computação
Centro Universitário Franciscano
97010-491 – Santa Maria – RS

[3]Centro de Tecnologia – Departamento de Eletrônica e Computação
Universidade Federal de Santa Maria
97105-900 – Santa Maria – RS

{alexz,reiner.perozzo,guilhermekurtz}@unifra.br

librelotto@inf.ufsm.br, solange.fagan@gmail.com

Abstract. *This paper aims to approach the cognitive agents from BDI theory to the area pervasive computing. This approximation will be given by mapping the behavioral specification of these agents in the language AgentSpeak(L) in a pervasive environment. We present an example in the educational context, i.e, an architecture of intelligent agents embedded in a pervasive environment, which assume the roles multimedia projector, automated blinds, audio system and lighting system, and integration of these devices.*

Resumo. *Este artigo propõe aproximar a teoria BDI de agentes cognitivos, especificados e implementados em AgentSpeak(L) e em seu interpretador Jason, com a área de Computação Pervasiva, por meio de um mapeamento de especificações comportamentais desses agentes num ambiente pervasivo. Busca-se apresentar uma situação exemplo baseada em dinâmica educacional, ou seja, apresentar uma arquitetura contendo agentes inteligentes que sejam associados a um ambiente pervasivo com dispositivos como projetor multimídia, persianas automatizadas, sistema de iluminação e sistema de áudio de uma sala de aula, de forma que esses dispositivos seriam integrados gerando um ambiente inteligente.*

1. Introdução

A computação está intrínseca no cotidiano humano, seja em residências, hospitais, trânsito, escolas, etc. Caminha-se, certamente, para um mundo em que computadores, em um ambiente qualquer, estarão voltados para tarefas específicas, mas interagindo uns com os outros para realizar ações em prol de um usuário. Há fortes indícios que movimentos estão ocorrendo para transformar espaços físicos em espaços computacionais inteligentes, onde os computadores podem oferecer serviços aos usuários não importando o local e nem a hora.

Um ambiente educacional equipado com dispositivos espalhados numa sala de aula poderia conferir características pervasivas. Esses equipamentos serviriam para projeção de *slides* em sistema multimídia e anotações em um quadro branco eletrônico. O ambiente integraria esses dispositivos, oferecendo suporte aos professores e aos alunos em suas atividades de aula.

De acordo com alguns dos principais autores da área, [Weiser 1991] e [Saha and Mukherjee 2003], assume-se que a Computação Pervasiva é a computação disponível em qualquer lugar, a qualquer tempo, e que o usuário possa usar qualquer dispositivo para ter acesso ao seu ambiente computacional. Sabendo-se, também, que as características básicas de agentes inteligentes são autonomia, mobilidade e proatividade, acredita-se que isso possa motivar a aproximação dessas duas áreas. Aproximar a teoria BDI (*Belief*, *Desire*, *Intention*) de agentes e Sistemas Multiagentes cognitivos no contexto da Computação Pervasiva pode ser concretizada pela integração da linguagem de especificação AgentSpeak(L) e seu ambiente de interpretação Jason com uma situação exemplo de Computação Pervasiva.

As aplicações pervasivas devem ser proativas, isto é, precisam identificar o que o usuário deseja e providenciar ações devidas no momento correto. Na computação pervasiva, o sistema se utiliza do contexto formado pelas informações relacionadas ao ambiente, seja por meio de sensores, seja pela geração ou envio a partir de algum dispositivo. Para que aplicações pervasivas funcionem adequadamente, é essencial que percepção de contexto e execução de tarefas sejam características básicas [Helal et al. 2005]. Segundo [Pereira and Librelotto 2012], essa percepção do contexto e a capacidade de utilizá-lo para executar tarefas são o que caracterizam uma aplicação sensível ao contexto (*context-awareness*) e também um sistema multiagente, conforme [Bordini et al. 2007]. Dessa forma, em termos práticos, o contexto englobaria sistema de áudio, sistema de vídeo, persianas automatizadas, projetor multimídia, *smartphones*, entre outros, sendo cada um desses dispositivos representados por agentes inteligentes.

Assim, propõe-se neste trabalho uma forma de realizar o mapeamento de especificações comportamentais de agentes cognitivos baseados no modelo BDI presentes em dispositivos de um ambiente pervasivo, com a finalidade de estender os comportamentos inteligentes desses agentes no ambiente proposto. Esse modelo de agentes é bastante estudado e encontra-se consolidado, por isso acredita-se que seja possível a aproximação com a área de Computação Pervasiva.

Para auxiliar no entendimento do artigo, o texto foi dividido em seções. A Seção 2 aborda o contexto de Computação Pervasiva: noções básicas e requisitos para se projetar um ambiente pervasivo. A Seção 3 trata dos conceitos sobre Sistemas Multiagentes, a teoria BDI de agentes inteligentes e suas tecnologias de especificação e implementação. A Seção 4 apresenta alguns trabalhos relacionados. A Seção 5 mostra a ideia geral da proposta deste artigo, bem como um estudo de caso no contexto educacional. Finalmente, algumas considerações e as referências bibliográficas do trabalho.

2. Computação Pervasiva

A pesquisa na área de Computação Pervasiva trabalha a ideia de que Computação Ubíqua é a computação que se move para fora das estações de trabalho e computadores pessoais e torna-se pervasiva no cotidiano do ser humano. A Computação Pervasiva possui como

referência a obra de [Weiser 1991]. Marc Weiser escreveu há mais de vinte anos que, no futuro, computadores estariam presentes nos mais simples objetos: etiquetas de roupas, xícaras de café, interruptores de luz, canetas, etc, de forma invisível para o usuário.

Os termos Computação Pervasiva, Computação Ubíqua e Computação Móvel são, muitas vezes, utilizados como sinônimos em alguns textos, entretanto, em [Lyytinen and Yoo 2002] e [Soldatos et al. 2007], mostra-se que eles são conceitualmente diferentes. A Computação Ubíqua integra mobilidade em larga escala com a funcionalidade da Computação Pervasiva, que define a estrutura do ambiente[Soldatos et al. 2007]. Segundo [Lyytinen and Yoo 2002], a Computação Móvel baseia-se na capacidade de mover fisicamente serviços computacionais, ou seja, o computador torna-se um dispositivo sempre presente, possibilitando ao usuário utilizar serviços que um computador oferece independentemente de sua localização. Contudo, esse recurso possui limitações, pois um dispositivo não é capaz de obter, de forma flexível, informação sobre o contexto em que a computação ocorre e ajustá-la corretamente. Já a Computação Ubíqua aproveita-se dos avanços da Computação Móvel e da Computação Pervasiva. A Computação Ubíqua surge da necessidade de integrar mobilidade com a funcionalidade da Computação Pervasiva, isto é, qualquer dispositivo computacional em movimento com um usuário pode construir, dinamicamente, modelos computacionais dos ambientes e configurar seus serviços dependendo da necessidade [Araújo 2003].

Esses autores também acreditam que o potencial de aplicações da Computação Ubíqua é limitado apenas pela imaginação, ou seja, com a conectividade, monitoramento e a coordenação de dispositivos localizados em casas, edifícios e carros inteligentes, através de redes sem fio (locais ou longa distância), a aplicação seria, por exemplo, para o controle de temperatura, luzes e umidade de uma residência, ou aplicações colaborativas com suporte à mobilidade. Estudos em Computação Ubíqua estão sendo realizados por pesquisadores do mundo todo, que vão de protótipos de rede que provêem acesso básico a qualquer tipo de dispositivo sem fio de forma transparente, segura, com tratamento de contexto, compressão, entrega e apresentação de conteúdo multimídia, até a adaptação da aplicação e da apresentação multimídia aos dispositivos do usuário [Araújo 2003].

Finalmente, segundo [Helal et al. 2005], a operacionalização de um ambiente pervasivo depende basicamente de três requisitos fundamentais:

- um sistema de automação contendo sensores, atuadores, controladores e interfaces humano-computador;
- uma infraestrutura de comunicação de dados para troca de informações entre dispositivos e entre usuários desse ambiente;
- sistemas inteligentes, cientes de contexto, adaptáveis, evolucionários e capazes de atender as necessidades dos usuários em suas atividades diárias, oferecendo suporte inteligente.

3. Sistemas Multiagentes

A construção de ambientes inteligentes representados por entidades de software com comportamentos sofisticados, com intensa interação e compartilhamento de recursos e objetivos, faz com que Sistemas Multiagentes sejam aplicados em diferentes propósitos. Segundo [Bordini and Hübner 2005], um agente é um software que age em prol do usuário ou de outro programa, obedecendo a regras de relacionamento do ambiente em que está

inserido. Tais ações comportamentais são decididas quando melhor lhe convir. Essa ideia é que agentes não são somente invocados exclusivamente por uma tarefa, mas também ativados por decisão própria. Existem diversos tipos de agentes, entre eles os cognitivos, que podem ser baseados em estados mentais, com capacidade de raciocínio, ou seja, capazes de obedecer um plano, ou planos, de ações que os levam a um estado pretendido. Esse tipo de agente possui características particulares como [Bordini and Hübner 2005] e [Vieira et al. 2006]: autonomia funcional; encontra-se continuamente em funcionamento; é sociável, ou seja, possui capacidade de interação; o mecanismo de controle é deliberativo; possui memória; e as sociedades são formadas por poucos agentes.

As estruturas desses agentes são constituídas por componentes mentais, como crenças, desejos, capacidades, escolhas e compromissos. Dentro das arquiteturas baseadas em estados mentais, encontra-se a abordagem de BDI, que tem o nome atribuído aos estados mentais: crenças (*beliefs*), desejos (*desires*) e intenções (*intentions*). Essa arquitetura representa seus processos internos através desses estados mentais e define um mecanismo de controle que seleciona de maneira racional o curso das ações [Bordini et al. 2007].

O principal enfoque dos Sistemas Multiagentes é promover mecanismos que possibilitem a criação de sistemas computacionais a partir de entidades de softwares autônomas que interagem através de um ambiente compartilhado por todos agentes de uma sociedade, e sobre o qual os agentes atuam alterando seu estado [Vieira et al. 2006]. Entre as tecnologias existentes para especificar e implementar agentes BDI, há a linguagem de implementação, AgentSpeak(L) e o interpretador Jason que possibilita a comunicação entre agentes [Bordini and Hübner 2005].

3.1. A Linguagem AgentSpeak(L) e o interpretador Jason

De acordo com [Bordini and Hübner 2005] e [Vieira et al. 2006], em AgentSpeak(L), um agente corresponde à especificação de um conjunto de crenças que formarão a base de crenças iniciais e um conjunto de planos. A base crenças de um agente é formada pela coleção de átomos de crenças e literais de crença: um átomo de crença forma um predicado de primeira ordem na notação lógica usual; as literais de crença são formadas por átomos de crenças ou suas negações. Os planos fazem referência a ações básicas que um agente é capaz de executar em seu ambiente. Essas ações são definidas por atributos com símbolos predicativos especiais (símbolos de ação) usados para atingir ações de produtos predicados. Um plano é formado por um evento ativador (propósito do plano), seguido de uma conjunção de literais de crença que representam um 'contexto'. O contexto deve ser conseqüência lógica do conjunto de crenças do agente no momento em que o evento é selecionado pelo agente para o plano ser considerado 'aplicável'. O restante do plano é seqüência de ações básicas ou sub-objetivos que o agente deve atingir ou testar quando uma instância do plano é selecionada para execução [Bordini and Hübner 2005].

Jason é um interpretador para uma extensão da linguagem AgentSpeak(L), que possibilita comunicação entre agentes baseada na teoria de atos de fala, por meio de diretivas de comunicação. A ferramenta Jason é implementada em Java (multiplataforma) e disponibilizada como código aberto sob a licença GNU LGPL [Bordini et al. 2007]. Além de interpretar a linguagem AgentSpeak(L) original, essa ferramenta possui características, como por exemplo [Bordini and Hübner 2005]: tratamento de falha em planos;

suporte para o desenvolvimento de ambientes (o ambiente é programado em Java); possibilidade de executar o SMA distribuidamente em uma rede; possibilidade de especializar em Java as funções de seleção de planos, e toda a arquitetura do agente; possui um ambiente de desenvolvimento, utilizando o jEdit (editor de texto de código aberto), ou em forma de um *plugin* para a IDE Eclipse.

4. Trabalhos correlatos

Em relação aos trabalhos correlacionados, foram encontrados trabalhos que de alguma forma propuseram construções de ambientes pervasivos, ou que utilizaram tecnologias que possibilitariam o desenvolvimento desses ambientes.

No trabalho de [Gárate et al. 2005], foi proposto uma interface na forma de mordomo (como um agente) que faz a comunicação dos usuários com os dispositivos do ambiente pervasivo. Em [da Silva et al. 2008], foi apresentado o modelo OCtoPUS que realiza a categorização de perfis de usuários baseada em seus comportamentos em um ambiente ubíquo. O modelo proposto utiliza tecnologias de web semântica para representar o conhecimento dos usuários.

A proposta SemantiCore Mobile [Escobar et al. 2007] permite o desenvolvimento de aplicações multiagentes em dispositivos móveis. Nela, foi abordado o uso de agentes e suas características em dispositivos móveis por meio do *framework* SemantiCore. Já [Wolski 2009] introduziu localização na plataforma Semanticore para criação de aplicações pervasivas baseadas em agentes de software, apresentando uma proposta de extensão da plataforma para que gere aplicações pervasivas orientadas a agentes.

Finalmente, [Junior et al. 2012] constatou que aplicações colaborativas vêm sendo utilizadas na integração de dispositivos computacionais em sala de aula. Entretanto, essa implementação expõem o programador a problemas de nível de sistema, tais como comunicação e concorrência. Dessa forma, esse trabalho buscou tornar os problemas transparentes ao desenvolvedor através de primitivas de compartilhamento de conteúdo, por meio de definições de operações de movimentação de conteúdo, tais como, mover, clonar e espelhar, que servem como um modelo de comunicação de alto nível para a construção dessas aplicações.

As obras citadas, de certa forma, trabalham com a ideia de agentes inteligentes ou classificação de perfis para a construção de ambientes pervasivos. Em [Wolski 2009], o autor afirma que existem várias plataformas para o desenvolvimento de sistemas multiagentes (SMAs) como JADE, Jason e SemantiCore e, nenhuma delas possui características para geração de aplicações pervasivas orientadas a agentes. Por essa outra situação, surge a motivação da realização deste estudo. Além do que, nenhum desses trabalhos utilizou as tecnologias AgentSpeak(L) e Jason.

5. Mapeamento de Agentspeak(L) para ambientes pervasivos: estudo de caso

Um ambiente educacional com características pervasivas poderia, por exemplo, ser composto por alguns dispositivos inseridos numa sala de aula, como para projeção de slides em sistema multimídia, anotações em um quadro branco eletrônico, o uso ou não de microfone e caixas de som. Esse ambiente seria formado por sistemas de iluminação e áudio, de projeção multimídia, de controle de persianas, de posicionamento (levantamento ou

rebaixamento) da tela de projeção e/ou quadro branco e de captação e distribuição de material produzido pelo professor (*slides*, textos, vídeos, etc). A integração desses sistemas daria suporte aos professores em suas atividades de reprodução de material didático e aos alunos na anotação das aulas de forma personalizada.

Numa visualização empírica dessa aplicação pervasiva, sempre que o professor utilizasse o quadro branco eletrônico, o sistema registraria os desenhos feitos e disponibilizaria essa informação sob a forma de hiperdocumentos multimídia sincronizados aos alunos conectados presentes na sala. Além disso, quando o professor chegasse em sala, com sua apresentação armazenada em seu *smartphone*, o ambiente se adequaria à exibição desse material, ou seja, as persianas e a tela de projeção seriam baixadas (e o quadro branco levantado), o sistema multimídia receberia o arquivo contendo a apresentação e entraria em modo de projeção, o sistema de áudio entraria em prontidão. Por fim, o professor poderia configurar o arquivo da apresentação para que fosse enviado, imediatamente, a todos os alunos em sala que estivessem conectados.

Sendo assim, faz-se necessário a descrição da arquitetura conceitual desse ambiente, constituída de seu *layout*, componentes, tecnologias, entre outros.

5.1. Arquitetura proposta

A arquitetura está orientada ao gerenciamento de serviços autônomos em ambientes de Computação Pervasiva, sendo que os agentes fariam o gerenciamento de cada um desses serviços. A arquitetura é constituída pelos componentes que seguem, conforme ilustrado na Figura 1.

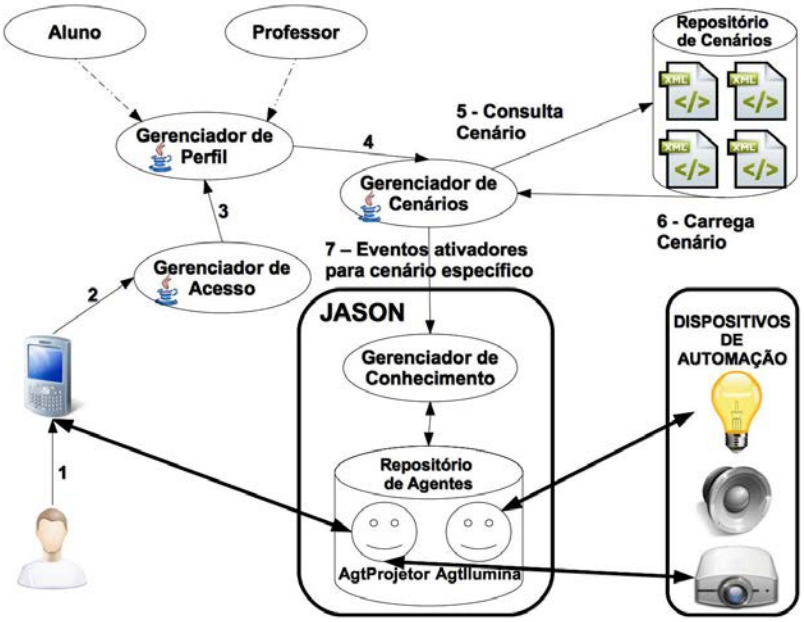

Figura 1. Ambiente Pervasivo proposto - conceitual.

- Gerenciador de Acesso: recebe e trata as conexões de acesso dos dispositivos móveis cadastrados;
- Gerenciador de Perfil: uma vez realizada a conexão, deve-se carregar o perfil do usuário conectado, seja ele aluno ou professor. Dependendo do perfil, haverá

recursos e permissões diferenciados, como por exemplo, enviar um arquivo ou acionar um dispositivo de automação;
- Gerenciador de Cenários: responsável por consultar e carregar os cenários do repositório de cenários conforme o perfil do usuário conectado, dos recursos e permissões associados a ele;
- Repositório de Cenários: cada cenário é um arquivo XML contendo a descrição de agentes e nome dos eventos ativadores (referentes ao dispositivo de automação específico);
- Gerenciador de Conhecimento: faz a relação dos cenários carregados e ações a serem realizadas pelos agentes;
- Repositório de Agentes: contém as implementações de todos os agentes (aspectos estruturais e funcionais) que representam os dispositivos de automação. Neste componente que a linguagem AgentSpeak(L) e o interpretador Jason se fazem presentes. A linguagem para especificar os comportamentos dos agentes, baseados em planos ativados por eventos externos ou eventos internos (de comunicação entre os agentes), tudo isso operacionalizado e integrado pelo interpretador Jason;
- Dispositivos de Automação: lâmpadas, projetor multimídia, caixas de som, persiana, quadro branco, etc.

Por fim, destacam-se, neste trabalho, os componentes Gerenciador de Cenários e Gerenciador de Conhecimento, isso porque os eventos serão disparados pelo Gerenciador de Cenários (carregados do Repositório de Cenários) para o Gerenciador de Conhecimento que se relaciona com o Repositório de Agentes. O agente conhece o protocolo de comunicação (via Jason) e implementa a interface de acesso ao Dispositivo de Automação.

5.2. Tecnologias

Em [Araújo 2003], é feita uma análise sobre dispositivos essenciais para a construção de ambientes pervasivos. Entre os citados, destacam-se controles, sensores e atuadores para residências e automóveis, eletrodomésticos, ar-condicionado, aquecedor, relógios e etiquetas inteligentes, além de toda a linha branca de utensílios domésticos, TVs, celulares, PDAs, consoles de jogos e muito mais. Esses dispositivos são categorizados como controles inteligentes, utensílios inteligentes, dispositivos de acesso à informação ou sistemas de entretenimento. Neste trabalho foca-se em controles inteligentes, que, por serem muito pequenos, podem ser integrados a lâmpadas, interruptores, sensores e atuadores. Essas aplicações variam de sensores (por exemplo, em portas para detectar a entrada de pessoas e seus *smartphones*) e atuadores (tais como, para acender/apagar lâmpadas específicas) à controle de projeção (sensores e atuadores para ligar/desligar/programar os sistemas de multimídia e áudio da sala de aula). Os controles são conectados a redes domésticas e gerenciados local ou remotamente.

Em relação à implementação de aplicações para ambientes pervasivos, há uma relação limitada de linguagens de programação disponíveis, como Java, C e C++, além de algumas linguagens proprietárias. A linguagem Java oferece independência de plataforma do código compilado e atende aos requisitos exigidos por dispositivos pequenos.

Os dispositivos de um sistema pervasivo processam os códigos executáveis e seus respectivos dados direto de sua localização em memórias RAM ou ROM. Ao desenvolver aplicações para dispositivos nesses ambientes, os programadores devem levar em

consideração algumas restrições, como capacidade limitada de entrada de dados, processamento limitado, memória, armazenamento persistente e vida da bateria, alta latência, largura de banda limitada e conectividade intermitente. A solução estratégica para o desenvolvimento de aplicações para dispositivos pequenos é remover características desnecessárias. E, se possível, tornar essas características como uma aplicação secundária e separada, pois aplicações menores usam menos memória no dispositivo e requerem menos tempo de instalação. Além disso, mover as tarefas de maior processamento e consumo para o servidor, permite que o dispositivo móvel trate a interface e utilize o mínimo de computação.

Dessa forma, a tecnologia Java seria uma opção para implementar o sistema pervasivo, principalmente por meio do interpretador Jason, que, como já mencionado, possibilita o desenvolvimento de ambientes, executa o sistema multiagente distribuidamente em uma rede e específica na linguagem Java as funções de seleção de planos e a arquitetura do agente.

5.3. Especificação AgentSpeak(L)

Para a modelagem ou especificação do Sistema Multiagente proposto, utilizou-se a metodologia Prometheus, que esta dividida em três etapas. A primeira etapa, tida como especificação do sistema, concentra-se em identificar os objetivos e as funcionalidades básicas do sistema, além das entradas (percepções) e saídas (ações). A segunda, Projeto Arquitetural, utiliza as saídas da etapa anterior para determinar quais os tipos de agentes que o sistema irá conter, e como eles irão interagir entre si. E a última, Projeto Detalhado, tem como objetivo modelar e detalhar a arquitetura interna dos agentes, e como eles deverão cumprir as suas tarefas [Padgham and Winikoff 2004].

Na Figura 2 é possível visualizar a legenda com os elementos básicos da diagramação em Prometheus, como atores, agentes, cenários, objetivos, planos, percepções, mensagens, base de dados e protocolo de comunicação.

Figura 2. Legenda Prometheus.

Na fase de especificação do sistema, assume-se dois principais objetivos do Sistema Multiagente, que são ”Professor chega na sala”e ”Professor chega na sala com arquivo para apresentação”, além dos subobjetivos atrelados. Na Figura 3 é possível observar as percepções ”Chegada de professor”e ”Chegada de arquivo marcado para apresentação”, que são eventos ativadores das ações como ”Ligar luzes para projeção”, ”Baixar cortinas”, entre outras, todas elas vinculadas aos papéis ”Prepara Sala de Projeção”, ”Distribui Arquivo”e ”Prepara Sala Tradicional”.

Finalmente, na Figura 4, é possível observar os agentes assumindo os papéis do sistema multiagente, percebendo ou sensorando o ambiente e disparando mensagens e/ou planos para que a sala entre ou não em configuração de apresentação, por exemplo.

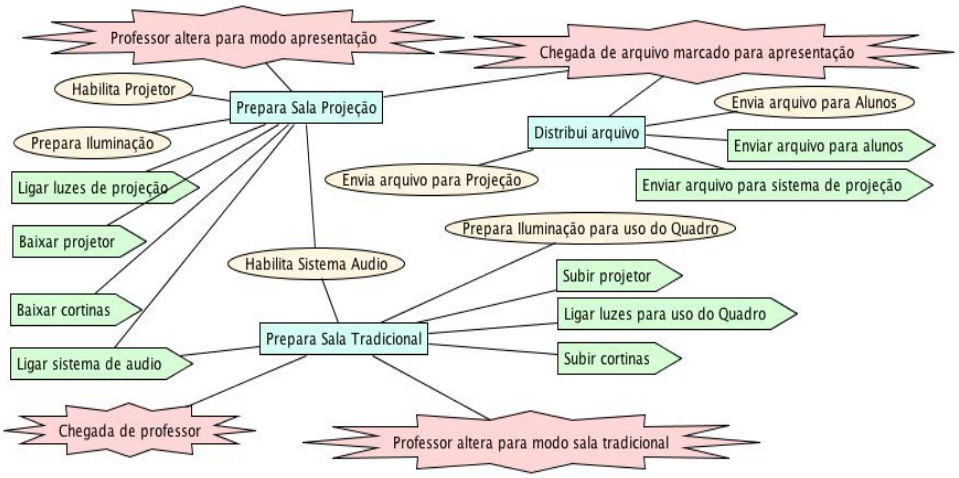

Figura 3. Papéis do Sistema Multiagente.

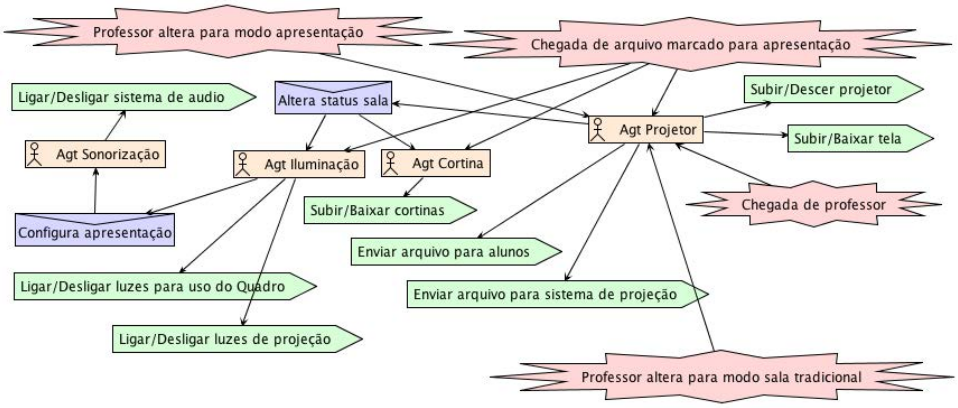

Figura 4. Visão geral do SMA.

5.4. Implementação Jason

Uma vez a modelagem estruturada, parte-se para a implementação do Sistema Multiagente por meio do interpretador Jason. Define-se, então, o projeto do sistema multiagente, como se pode visualizar na Figura 5. No arquivo (.mas2j), definem-se a infraestrutura (centralizada ou distribuída) de funcionamento, o ambiente (.java) e todos os agentes envolvidos no ambiente.

Na Figura 6 é possível observar uma implementação de comportamento do agente ”agtProjetor”, em que há um plano que trata o evento externo ”professor(emSala)”, gerado pelo ambiente. Esse plano só é disparado caso exista o contexto ou a condição para isso, que são as crenças presentes no agente ”agtProjeto”sobre a sala estar livre e o professor estar com arquivo da apresentação. Uma vez o plano tendo contexto, disparam-se ações que devem ser executadas no ambiente, como ”enviarArquivoParaAlunos”e ”enviarArquivoParaProjecao”. Uma atualização na base de crença também é executada, ou seja, retirada da crença ”statusSala(livre)”e adição de nova crença ”statusSala(ocupado)”. Além disso, uma mensagem é encaminhada a todos os agentes para que executem o plano ”ambiente(paraApresentacao)”.

As ações encaminhadas para execução no ambiente são implementadas no arquivo

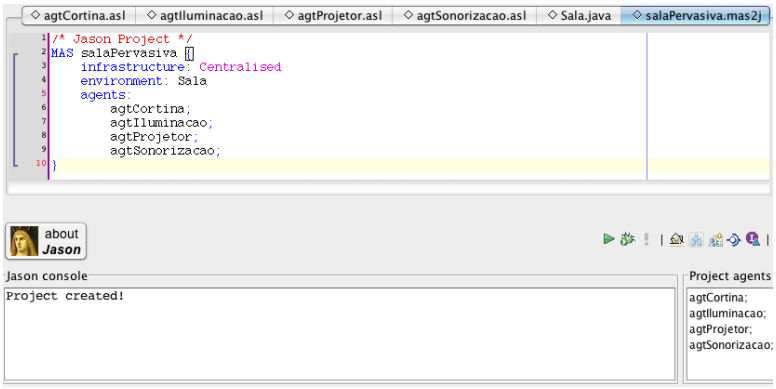

Figura 5. SMA em Jason da Sala Pervasiva.

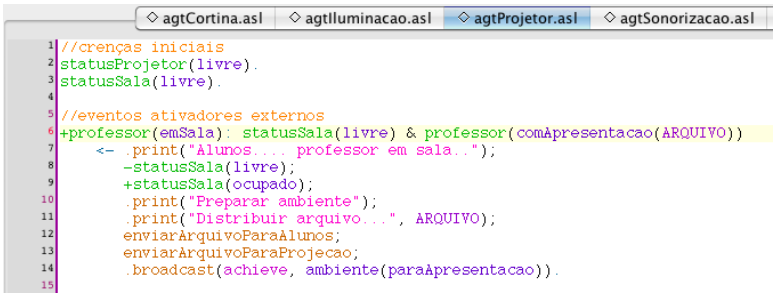

Figura 6. Implementação dos comportamentos do agente projetor.

”Sala.java”, que é o ambiente deste SMA. Na Figura 7, é possível visualizar o método ”*executeAction()*”que trata as ações encaminhadas pelos agentes: ”enviarArquivoPara-Projecao”e ”enviarArquivoParaAlunos”.

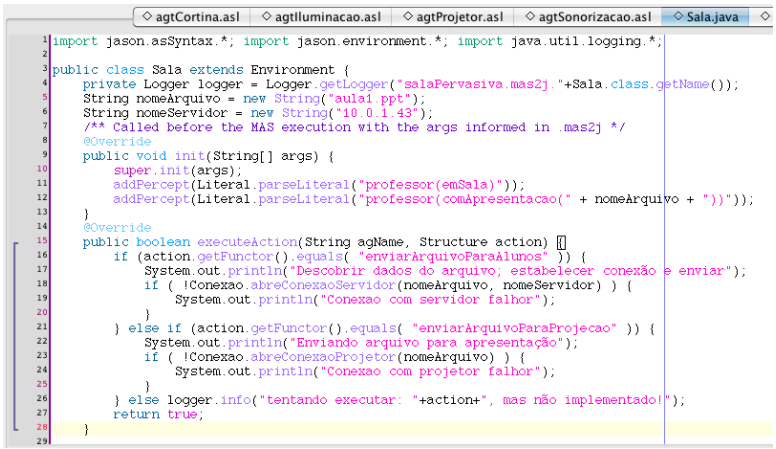

Figura 7. Implementação das ações pelo ambiente.

6. Considerações Finais

Ao longo do texto, buscou-se discutir a Computação Pervasiva relacionada a Sistemas Multiagentes, unindo o melhor das duas áreas. Enquanto a pervasividade agrega a habilidade de perceber o ambiente, agentes somam quanto a autonomia e proatividade. Porém,

ambas possuem o caráter de sistemas reativos (reagindo a eventos no ambiente), flexibilidade, adaptabilidade e mobilidade. Além disso, são capazes de gerar comunicação entre seus elementos. Por isso, acredita-se, com este estudo, que a área de Sistemas Multiagentes pode ser um meio eficaz de operacionalizar a Computação Pervasiva. Acredita-se, também, que o uso da teoria BDI, por meio da linguagem AgentSpeak(L) e seu interpretador Jason, foi fundamental para a integração dessas duas áreas, uma vez que o Jason fornece recursos de comunicação e concorrência entre os agentes.

A área de Sistemas Multiagentes encontra-se consolidada, com metodologias e ferramentas para projeção e implementação de sistemas inteligentes. Entre as diversas tecnologias existentes, destacam-se a linguagem AgentSpeak(L) e o interpretador Jason que, de fato, possibilitam a concretização de sistemas desse tipo. O interpretador, por basear-se na linguagem Java, auxilia sobremaneira quanto a portabilidade e integração dos componentes que compõem o sistema pervasivo, por ser interpretada, é executada nos mais diversos equipamentos de forma distribuída, inclusive.

Como trabalhos futuros, propõe-se um estudo experimental quantitativo para avaliar o desempenho do sistema multiagente construído. Também, está previsto a implementação da integração dos demais processos de gerenciamento da arquitetura proposta para o estudo de caso.

Finalmente, registra-se que com este estudo foi possível verificar a aproximação dessas áreas.

Referências

Araújo, R. B. (2003). Computação ubíqua: Princípios, tecnologias e desafios. In *Computação Ubíqua: Princípios, Tecnologias e Desafios*, volume 1, pages 1–71. SBC, 1 edition.

Bordini, R. H. and Hübner, J. F. (2005). BDI agent programming in agentspeak using Jason (tutorial paper). In *CLIMA*, pages 143–164.

Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming multi-agent systems in AgentSpeak using Jason*. Wiley.

da Silva, D. M., Barbosa, J., and Vieira, R. (2008). OCtoPUS: um modelo para classificação de perfil de usuários usando trilhas em sistemas ubíquos. In *Companion Proceedings of the XIV Brazilian Symposium on Multimedia and the Web*, WebMedia '08, pages 185–188, New York, NY, USA. ACM.

Escobar, M. S., Ries, L., P., L. A., and M., B. (2007). Semanticore mobile - permitindo o desenvolvimento de aplicações multiagentes em dispositivos móveis. In *I Workshop on Pervasive and Ubiquitous Computing (WPUC)*.

Gárate, A., Herrasti, N., and López, A. (2005). GENIO: an ambient intelligence application in home automation and entertainment environment. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, sOc-EUSAI '05, pages 241–245, New York, NY, USA. ACM.

Helal, S., Mann, W., El-Zabadani, H., King, J., Kaddoura, Y., and Jansen, E. (2005). The gator tech smart house: A programmable pervasive space. *Computer*, 38(3):50–60.

Junior, M. R., Freitas, L., Massarani, M. A., da Rocha, R., and Costa, F. (2012). Ucle: Um middleware de computação ubíqua para compartilhamento de conteúdo em salas de aula inteligentes. In *SBCUP - IV Simpósio Brasileiro de Computação Ubíqua e Pervasiva*. SBC.

Lyytinen, K. and Yoo, Y. (2002). Issues and challenges in ubiquitous computing. *Commun. ACM*, 45(12):62–65.

Mohsin, M. and Ahmad, A. (2014). Genetically-encoded nanosensor for quantitative monitoring of methionine in bacterial and yeast cells. *Biosensors and Bioelectronics*, 59:358–364. cited By (since 1996)0.

Padgham, L. and Winikoff, M. (2004). *Developing Intelligent Agent Systems: A practical guide.* John Wiley & Sons.

Pereira, H. and Librelotto, G. (2012). *ARCP: uma arquitetura para a utilização de computação nas nuvens nos ambientes de computação pervasiva.* Amazon.

Saha, D. and Mukherjee, A. (2003). Pervasive computing: a paradigm for the 21st century. *Computer*, 36(3):25–31.

Soldatos, J., Dimakis, N., Stamatis, K., and Polymenakos, L. (2007). A breadboard architecture for pervasive context-aware services in smart spaces: middleware components and prototype applications. *Personal Ubiquitous Comput.*, 11:193–212.

Vieira, R., Moreira, Á. F., Bordini, R. H., and Hübner, J. F. (2006). An agent-oriented programming language for computing in context. In *IFIP PPAI*, pages 61–70.

Weiser, M. (1991). The computer for the 21st century. In Baecker, R. M., Grudin, J., Buxton, W. A. S., and Greenberg, S., editors, *Human-computer interaction*, pages 933–940. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Wolski, L. Z. (2009). Introduzindo localização na plataforma semanticore para criação de aplicações pervasivas baseadas em agentes de software. Master's thesis, PUCRS.