

YLANA KIPUNA DOS SANTOS FIGUEIREDO

Orientador: Fernando Cortez Sica

**MODELAGEM DE FERRAMENTAS FOCADAS EM
ONTOLOGIAS PARA A EAD BASEADAS EM TEORIAS
SOCIAIS E AGENTES INTELIGENTES**

Ouro Preto
Novembro de 2010

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**MODELAGEM DE FERRAMENTAS FOCADAS EM
ONTOLOGIAS PARA A EAD BASEADAS EM TEORIAS
SOCIAIS E AGENTES INTELIGENTES**

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

YLANA KIPUNA DOS SANTOS FIGUEIREDO

Ouro Preto
Novembro de 2010



UNIVERSIDADE FEDERAL DE OURO PRETO

FOLHA DE APROVAÇÃO

Modelagem de ferramentas focadas em ontologias para a EaD baseadas
em teorias sociais e agentes inteligentes

YLANA KIPUNA DOS SANTOS FIGUEIREDO

Monografia defendida e aprovada pela banca examinadora constituída por:

Dr. FERNANDO CORTEZ SICA – Orientador
Universidade Federal de Ouro Preto

Dr. CARLOS ALBERTO MARQUES PIETROBON
Universidade Federal de Ouro Preto

Dr. JOSÉ ROMILDO MALAQUIAS
Universidade Federal de Ouro Preto

Ouro Preto, Novembro de 2010

Resumo

Este trabalho demonstra o uso de ontologias baseadas em perfis sociais para o processo de criação de ferramentas computacionais a serem utilizadas pela EaD (Educação a Distância). Com a contextualização da ontologia baseada em perfis sociais, é permissível a modelagem de agentes inteligentes aplicados à educação. Neste caso, os agentes comportam-se como proxies dos cursistas afim de sejam estabelecidas interações ou comunicações entre os demais agentes pertencentes à comunidade e, também, afim de que sejam inspecionadas as atividades dos cursistas inseridos em um grupo (comunidade) específica. Estes agentes são criados levando-se em conta o conceito de grupos modelados a partir de uma ontologia baseada em perfis sociais, tendo como foco "interesses comuns" entre os integrantes do grupo.

Palavras-chave: Ontologia.Perfis Sociais. EaD. Agentes Inteligentes

Abstract

This work demonstrates the use of ontologies based on social profiles for the process of creating computational tools to be used by DE (Distance Education). In the context of ontology based on social profiles, it is permissible for modeling intelligent agents for education.

In this case, the agents act as proxies of the course participants in order to be established communications or interactions between other agents belonging to the community and also be inspected in order that the activities of the course participants within a group (community) specific. These agents are created taking into account the concept of groups modeled after an ontology based on social profiles, "focusing on common interests" among members of the group.

Keywords: Ontology. Social Profiles. DE. Intelligent Agents.

Dedico este trabalho aos meus pais, que sempre me apoiaram em todos os momentos, que sempre me deram uma palavra de conforto e carinho nas horas mais conflitantes da minha vida e muitas vezes sem mesmo saberem. E sempre fizeram tudo para me dar uma boa educação e para que nada me faltasse me ajudando a alcançar mais uma meta.

Agradecimentos

Agradeço ao Senhor, pois sem Ele nada sou, sem Ele nada teria sido realizado e não consegueria alcançar mais uma etapa da minha vida. Senhor, tudo o que sou e o tenho devo a ti, pois sempre me amparaste em todas as horas, nos momentos mais difíceis da minha vida, me deste forças para me re-erguer, levantar a cabeça e batalhar, me deste discernimento para superar os obstáculos e alcançar os meus objetivos. E mesmo não sendo uma boa filha, mesmo comentendo faltas, a tua mão sempre segura a minha, sempre estiveste presente, sempre foste o meu o Melhor Amigo. Muito obrigado, meu Deus!

Aos meus queridos papás, Humberto e Teresa, pelo amor imensurável que têm por mim, por toda dedicação, por todos os ensinamentos, por todos os conselhos, por todas as palavras de consolo e principalmente, por acreditarem em mim, acreditarem que a minha vinda ao Brasil seria para realizar o objetivo de me formar e me transformar na pessoa que hoje sou.

As minhas avós, Joana e Aurora e a minha tia do Coração, Guerrita.

Ao Osvaldo, pela amizade sincera, pelas palavras de conforto e encorajamento nos momentos de incerteza, por todas as tristezas e alegrias divididas, por todo carinho e atenção doados nos momentos em que precisei e por todos os seus bons conselhos nas horas certas.

As minhas amigas, que sempre estarão no meu coração, Paloma, Âurea, Nelma, Suelaine e Sílvia por todos momentos de alegrias e tristezas compartilhados.

Aos meus colegas e amigos de graduação, em especial Daniel, Igor, Pablo, Rodrigo, Marcos, por dividirem comigo alguns obstáculos e por toda a força e amizade depositadas.

Ao meu Orientador e Professor, Sica, por toda ajuda, por toda orientação, durante a realização deste trabalho, pelas suas palavras amigas nos momentos de tensão e por todos os seus ensinamentos.

A todos os mestres que contribuíram que para minha formação acadêmica e engrandecimento como instrumento da sociedade.

A Ouro Preto, que me ensinou a crescer e a todos àqueles que fizeram parte da minha jornada.

Sumário

1	Introdução	1
2	Justificativa	3
3	Objetivos	4
3.1	Objetivo Geral	4
3.2	Objetivos Específicos	4
4	Contextualização	5
5	Ontologia	9
5.1	Definição	9
5.2	Vantagens	11
5.3	Componentes de um Ontologia	11
5.4	Classificação das ontologias	12
5.5	Principais áreas de aplicação das ontologias	13
5.6	Abordagens para o desenvolvimento de uma ontologia	14
5.7	Metodologias para desenvolvimento de uma ontologia	15
5.8	Metodologia para criação de uma ontologia	16
5.9	Processos de aprendizagem baseados na interação social	18
5.9.1	Desenvolvimento em uma perspectiva social	19
5.10	Ontologia para aluno e grupo de alunos	20
6	Agentes Inteligentes e a Educação	22
6.1	Agentes Inteligentes	23
6.2	Categoria de Agentes	24
6.3	Aplicações dos Agentes	26
6.4	Sistemas Multiagentes	27
6.5	Comunicação entre agentes	28
6.5.1	Comunicação direta	28
6.5.2	Comunicação por sistemas federados	28

6.5.3	Comunicação por broadcast	29
6.5.4	Comunicação por blackboard	30
6.6	Protocolos de comunicação entre agentes	30
7	FIPA	32
7.1	Parâmetros das mensagens do FIPA-ACL	32
7.2	Especificações FIPA	32
7.2.1	Aplicações de Sistemas Multiagentes	33
7.2.2	Arquitetura Abstrata para Sistemas Multiagentes	33
7.2.3	Comunicação entre Agentes	33
7.2.4	Gerenciamento de Agentes	35
7.2.5	Transporte de Mensagens entre Agentes	35
8	JADE	37
8.1	Características do JADE	38
8.2	Regras de comunicação FIPA no JADE	39
8.3	Agentes em JADE	40
8.3.1	Ciclo de vida do Agente	41
8.4	Troca de Mensagens	42
8.5	Interoperabilidade	42
9	Implementação	44
10	Conclusões	45
	Referências Bibliográficas	48

Lista de Figuras

5.1	Exemplo de ontologia de domínio de circuitos eletrônicos	10
5.2	Exemplo da base de conhecimento que utiliza a ontologia de circuitos eletrônicos .	10
5.3	Diferentes tipos de ontologias e suas relações	13
6.1	Uso do computador na EaD	23
6.2	Esquema de agente	25
6.3	Comunicação Direta entre agentes	29
6.4	Comunicação por sistema federado ou comunicação assistida	29
6.5	Comunicação por quadro negro	30
8.1	Modelo da plataforma de agentes definido pela FIPA	39
8.2	Arquitetura interna de uma agente genérico em JADE	41
8.3	Ciclo de Vida de um agente definido pela FIPA	42
8.4	Interoperabilidade entre os agentes	43

Lista de Tabelas

7.1	Parâmetros das mensagens FIPA-ACL	34
7.2	Especificações FIPA: Aplicações de Sistemas Multiagentes	34
7.3	Especificações FIPA: Arquitetura Abstrata	34
7.4	Especificações FIPA: Gerenciamento de agentes	35
7.5	Especificações FIPA: Transporte de Mensagens entre Agentes	36

Lista de Algoritmos

Capítulo 1

Introdução

Nas últimas décadas, tem visto uma grande massa de cidadãos com a preocupação de terem uma educação e para atender a esta grande massa, evidenciou-se a importância da educação a distância (*EaD*), realizada a princípio por meio de correspondência, posteriormente através do uso de meios de comunicação como o rádio e a televisão associados a materiais impressos enviados pelo correio. O advento das tecnologias de informação e comunicação trouxe novas perspectivas para a educação a distância, levando universidades, escolas, centros de ensino, organizações empresariais e grupos de profissionais de educação a se dedicarem ao desenvolvimento de cursos a distância com suporte em ambientes digitais de aprendizagem acessados via internet, os quais assumem distintas abordagens. Este trabalho tem por objetivo desenvolver uma ferramenta que possa facilitar o estudo e aprendizado do aluno em um ambiente virtual para educação a distância. Com base nisso, pretende-se criar uma ontologia que propicie uma base para o desenvolvimento do trabalho proposto para esta monografia.

Para que os pontos inerentes ao desenvolvimento do trabalho sejam conhecidos, esta monografia inicia-se apresentando com uma breve introdução sobre o trabalho proposto, posteriormente, no capítulo 2 é apresentada a justificativa do trabalho dando ênfase a sua relevância.

No capítulo 3 são apresentados os objetivos gerais e específicos, ajudando a compreender melhor a relevância deste trabalho.

O capítulo 4 apresenta a contextualização da proposta de trabalho, que tem como base regras e ações sociais que partem de teorias de pensadores modernos e sociólogos como Durkheim e Vygotsky. Através destas teorias sociais é possível ter-se a idéia de como podem se pode criar grupos que atendem um certo objetivo, estes grupos serão formados por alunos, uma vez que o foco deste trabalho é a criação de uma ferramenta para educação a distância baseada em uma ontologia de aluno, que visa a interação entre alunos de um certo curso, através de grupos constituídos por eles.

Como citado no parágrafo anterior a modelagem dos alunos e dos grupos será feita através da criação de ontologias para aluno e grupo, respectivamente, que a posterior será utilizada para a implementação dos agentes inteligentes através de um framework do Java apropriado

para este tipo de aplicação, uma vez que este já possui todas regras para o estabelecimento da comunicação entre agentes, nesse caso, os alunos. As regras de comunicação são estabelecidas pelos protocolos do FIPA, usados pelo JADE.

O capítulo 5 apresenta o conceito de Ontologia, suas características e relações, de seguida, no capítulo 6 são apresentados alguns conceitos de Inteligência Artificial dando ênfase à área de sistemas multiagentes e suas principais características.

No sub capítulo 7 será feita uma breve apresentação sobre as especificações produzidas pela fundação FIPA que cria protocolos que facilitam a comunicação entre agentes. O capítulo 8 falará sobre o framework JADE adoptado para implementação do sistema multiagente proposto no trabalho, bem como algumas das suas especificações e suas diversas aplicações.

O capítulo 9 trata da implementação do trabalho, onde será feita uma breve descrição do código de implementação dos agentes inteligentes, modelados através de uma ontologia. Também serão apresentados os códigos das classes implementadas e algumas informações inerentes as classes serão acrescentadas.

No capítulo 10 serão apresentados os resultados obtidos e por fim, no capítulo 11 serão apresentadas as conclusões obtidas durante a concepção do trabalho.

Capítulo 2

Justificativa

Este trabalho se justifica pela relevância e contribuição apresentada pela fusão de teorias sociais e informática. Justifica-se, ainda, pela importância que os ambientes virtuais de aprendizagem e a própria EaD vem tomando no cenário educacional e social. Cenário representado por iniciativas privadas e públicas, como por exemplo, o programa Proinfo do governo federal.

Diante da crescente demanda dos ambientes e ferramentas relacionados à EaD, espera-se, com este trabalho, contribuir para que novas ferramentas e metodologias possam ser criadas e consolidadas de forma a estabelecer novos caminhos e horizontes para os processos de ensino-aprendizagem.

Aliado ao fato da importância da representação ontológica no cenário computacional, tem-se, também, a relevância da Educação a Distância EaD no contexto da educação, havendo neste campo a participação em massa da computação representada pelas NTICs Novas Tecnologias de Interação e Comunicação, dentre as quais incluem-se os AVAs Ambientes Virtuais de Aprendizagem.

Devido às importâncias e relevâncias apresentadas, pretende-se com esta monografia, focar o uso das ontologias no processo de criação de ferramentas computacionais para a EaD de forma a consolidar perfis de ferramentas baseadas em perfis sociais.

Capítulo 3

Objetivos

Nesta seção serão apresentados os objetivos gerais e específicos do trabalho.

3.1 Objetivo Geral

Desenvolver uma ferramenta para EaD, partindo da criação de uma ontologia baseada em perfis sociais de alunos e agentes inteligentes modelados por meio de um framework do Java, apropriado para o desenvolvimento aplicações orientadas a agentes e interação entre eles, o JADE (Java Agent Development framework), que é um ambiente que propõe toda uma infraestrutura de suporte e desenvolvimento de sistemas multiagentes, tornando a implementação mais simples e robusta.

3.2 Objetivos Específicos

- Estudar representações na forma de ontologias;
- Estudar pensadores da ciência social relacionados à educação;
- Estabelecer contextos e fronteiras entre a educação mediada por computador e os parâmetros sociais;
- Estudar conceitos inerentes aos agentes inteligentes aplicados à educação;
- Criar uma representação baseada em ontologia;
- Criar de metodologias de ensino e agentes inteligentes;
- Implementar mecanismos de interação inter-agentes, com o uso da ferramenta JADE.

Capítulo 4

Contextualização

A partir da contextualização das regras e ações sociais baseadas em teorias de pensadores modernos, como Durkheim, Vygotsky e outros sob a forma de ontologia e do estabelecimento de relações das teorias sociais com o processo de ensino-aprendizagem mediada por computador, pretende-se modelar ferramentas computacionais para a EaD. Esta contextualização será utilizada para alimentar as regras de interação, supervisão, controle e mediação dos agentes inteligentes modelados como proxies dos cursistas. Entende-se como "proxy agent" o modelo de agente modelado para inspecionar as atividades desenvolvidas pelos cursistas de modo a interagir com os demais agentes pertencentes à comunidade e propor fluxos de execução para as ações de ensino-aprendizagem.

Para estabelecer parâmetros e conceitos afim de facilitar o processo de criação de ontologias, é importante o conhecimento das teorias e contextualização criadas pelos pensadores e sociólogos. A seguir, são citados e retratados alguns pensadores de suma importância ao entendimento da atual conjuntura educacional.

Segundo Durkheim Allahverdi (1999), sociólogo da educação positivista:

"as pessoas têm incorporadas em si dois seres. O primeiro, é o ser individual que se caracteriza pelos estados mentais de cada um e pelos aspectos de sua vida pessoal. O segundo é o ser social, voltado para os comportamentos relacionados à sociedade em que vivemos."

Para Durkheim a educação deveria ao mesmo tempo, ter uma base comum e diversificada. Apesar das diferenças sociais todas as crianças devem receber idéias e práticas, que são valores do seu povo, da sua nação. Essa seria a base comum da educação, pois contem os conhecimentos que deveriam ser compartilhados por todos.

Durkheim não desenvolveu métodos pedagógicos, mas suas idéias ajudaram a compreender o significado social do trabalho do professor, tirando a educação escolar da perspectiva individualista, sempre limitada pelo psicologismo idealista (Revista Escola - Durkheim (2008)). Segundo Durkheim, o papel da ação educativa é formar um cidadão que tomará parte do espaço público. Não somente o desenvolvimento individual do aluno. A educação tem por

objetivo suscitar e desenvolver na criança estados físicos e morais que são requeridos pela sociedade política no seu conjunto. Tais exigências, com forte influência no processo de ensino, estão relacionadas à religião, às normas e sanções, à ação política, ao grau de desenvolvimento das ciências e até mesmo ao estado de progresso da indústria local (Revista Escola - Durkheim (2008)).

Já neste ponto, pode-se traçar paralelos dos princípios de Durkheim com a utilização de agentes inteligentes como agentes mediadores do processo de ensino-aprendizagem. Como se pôde notar, Durkheim tem como uma de suas sugestões básicas, a introdução de elementos que permitam a interação e a construção colaborativa e cooperativa do conhecimento. Sendo assim, a determinação de ontologias contextualizando ações sociais no âmbito educacional, junto com a implementação através da implementação de agentes inteligentes, vêm ao encontro dos princípios deste pensador.

Para além de Durkheim, outros sociólogos também tiveram uma participação extremamente importante no desenvolvimento de novas teorias educacionais, que hoje representam um papel muito importante não só na prática da educação presencial, como também na educação a distância. Dentre estes, destaca-se Vygotsky, sociólogo interacionista que elaborou sua teoria tendo por base o desenvolvimento do indivíduo como resultado de um processo sócio-histórico, destacando o papel da linguagem e da aprendizagem nesse desenvolvimento, passando a ser uma teoria histórico-social (Revista Escola - Vygotsky (2008)). Uma idéia principal para a compreensão das concepções sobre desenvolvimento humano como processo sócio-histórico é a idéia de mediação. Vygotsky enfatiza a construção do conhecimento como uma interação mediada por várias relações, seja elas intra ou interpessoais, ou seja, o conhecimento não está sendo visto como uma ação do sujeito sobre a realidade e sim, pela mediação feita por outros sujeitos (Vygotsky (2001)).

Portanto, para Vygotsky, a aquisição de conhecimentos se opera pela interação do sujeito com o meio, ou seja, o sujeito não é apenas ativo, mas interativo. Tendo em vista sua teoria, Vygotsky considerava o aluno não somente como um sujeito de aprendizagem, mas aquele que aprende junto ao outro o que o seu grupo social produz, tal como: valores, linguagem e o próprio conhecimento Daniels (2003) "apud" Vygotsky (2001)). A partir desta teoria e da teoria de Durkheim, tem-se a idéia de desenvolver os agentes inteligentes por meio de um framework específico do JAVA, modelando-os como grupos, sendo que cada grupo detém as próprias características sociais e os elementos que os constituem têm a capacidade de interagir uns com os outros e até mesmo com elementos não percentes ao grupo. Um exemplo de grupo seria detendo características a ele inerentes seria:

- Grupo para alunos que frequentassem uma determinada série.
- Grupo destinado a discussão e ao aperfeiçoamento do aprendizado em uma determinada disciplina, por exemplo, matemática.

- Estabelecendo-se a disciplina a ser estudada, determinar o grau de dificuldade dos alunos dentro da disciplina e criar formas mais fáceis de mediar os alunos. Neste caso há a necessidade de um mediador do grupo.
- Para alunos com dificuldades menos acentuadas, estes podiam enxergar o grupo como um repositório de material de estudos, como trabalhos resolvidos, listas de exercícios, provas, etc.
- Os próprios alunos podiam sanar as dúvidas uns dos outros, por meio de troca de mensagens.
- Alunos podem ser convidados a participar do grupo.

Após o estudo dessas teorias sociais voltadas para a educação que contribuíram para a modernização do sistema educacional, culminando em ambientes em que os próprios alunos se responsabilizam pelo processo individual de aprendizagem, pesquisas apontam que há necessidade de se criar ambientes que tenham a capacidade de se adaptarem a este contexto e de se personalizarem de acordo com as características dos alunos. Daí, mais uma vez, se pode ver a importância do conceito de grupos para a modelagem dos agentes inteligentes, ou seja, os próprios cursistas de formas a atender as suas necessidades. Isso, leva à introdução de técnicas de IA com a finalidade de propiciar mecanismos de modelagem do processo de ensino bem como do estado cognitivo do estudante. Cabe a partir deste ponto mostrar um pouco sobre agentes inteligentes, suas propriedades e funcionamento, entretanto, será definido um capítulo para aprofundar mais o estudo sobre agentes inteligentes.

Agentes inteligentes é um ramo da IA (*inteligência artificial*), constituído por componentes capazes de organizar, selecionar, produzir informações e tomar decisões a partir de alguma fonte de dados. Os agentes devem deter de algumas propriedades para que eles possam se comunicar e cooperar com outros agentes que coexistem no mesmo ambiente. São elas: **Autonomia, Reatividade, Pró-atividade, Intencionalidade, Racionalidade, Continuidade temporal, Sociabilidade, Benevolência, Adaptabilidade, Mobilidade**, que serão explicadas mais adiante.

Para se estabelecer a interação entre os agentes, é necessário que haja uma arquitetura que define processos que vão possibilitar a interação dos agentes. Além da arquitetura que facilita a comunicação entre agentes, criando assim, um sistema multiagente, ou seja, vários agentes acoplados dentro de um ambiente, é necessário que haja um meio de comunicação entre esses agentes, com protocolos bem definidos para o processo de troca de mensagens entre os agentes. Para isso, é necessário o uso de ferramentas apropriadas que modelam não os próprios agentes como também têm as regras que definem os protocolos de troca de mensagem. E a melhor ferramenta que existe atualmente no mercado é o framework do JAVA conhecido por JADE (*Java Agent Development framework*) como próprio nome diz, é um

framework específico para o desenvolvimento de agentes, ele trata da comunicação, do ciclo de vida e do monitoramento da execução dos agentes. É importante citar que a comunicação entre os agentes feita pelo JADE se baseia nos padrões de comunicação para tecnologias de sistemas multiagentes estabelecidos pela FIPA (*Foundation For Intelligent Physical Agents*), a partir disso é possível então estabelecer-se uma comunicação distribuída entre os agentes.

Mais detalhes sobre o JADE, FIPA e protocolos de comunicação, serão vistos em um capítulo específico que falará sobre o assunto.

Capítulo 5

Ontologia

É originária da filosofia, ela lida com a natureza e a organização do ser. Esse termo foi introduzido por Aristóteles em *Metafísica* (Puc-Rio (2003) "apud" Maedche (2002)).

A ontologia deve ser explícita, formal¹ e descrever um conhecimento comum a um grupo determinado. Estruturalmente, deve possuir um conjunto de termos organizados com uma hierarquia associada, ou seja, uma taxinomia que servirá de esquema para a base de conhecimentos.

Atualmente, vem se verificando o uso de ontologias em diversas áreas da informática, dentre elas destacam-se: o uso de ontologias para o processamento de linguagens naturais (Lopes (2005) "apud" Moreno e Hernández (2000)); na área de e-commerce CITAÇÃO; gerenciamento do conhecimento CITAÇÃO; web semântica (Lopes (2005) "apud" Maedche (2002)) e na área de educação (Severo et al. (2009) "apud" Maedche (2002)), que é o foco deste trabalho.

5.1 Definição

A literatura sobre ontologias apresenta uma série de definições distintas com pontos de vista diferentes e até mesmo complementares para uma mesma realidade. Segundo o artigo sobre ontologia da PUC-RIO, Guarino faz uma longa discussão sobre o significado preciso do termo dentro da Ciência da Computação, pois o seu significado tende a variar conforme o objetivo do uso da ontologia.

Uma ontologia pode ser definida como algo que fornece um conjunto de conceitos e termos para descrever um determinado domínio. Sendo que os termos definidos servem para descrever uma determinada realidade, que serão usados por uma base de conhecimentos, porém a ontologia permanece inalterada, desde que o domínio se mantenha inalterado.

Uma outra definição de ontologia, seria a definição proposta por Fensel (2001) proveniente do artigo editado pela PUC-RIO :

¹passível de ser processada por uma máquina

” Uma ontologia é uma especificação formal explícita de uma conceitualização compartilhada.”

Pode-se usar os mesmos conceitos descritos na ontologia para descrever diferentes realidades na base conhecimentos.

A ontologia possui axiomas, ou seja, regras pertinentes ao domínio em questão.

A seguir, é apresentado um exemplo ilustrativo para diferenciar uma ontologia de uma base de conhecimentos. Neste exemplo, é definida uma ontologia para o domínio de circuitos eletrônicos, que incluiria conceitos específicos (por exemplo: circuito integrado, resistor, transistor processador, diodo, etc) e também relações entre esses conceito (por exemplo: o processador é um tipo de circuito integrado, etc) Puc-Rio (2003) e com isto podemos descrever uma base de conhecimentos que descreva o circuito de um rádio.

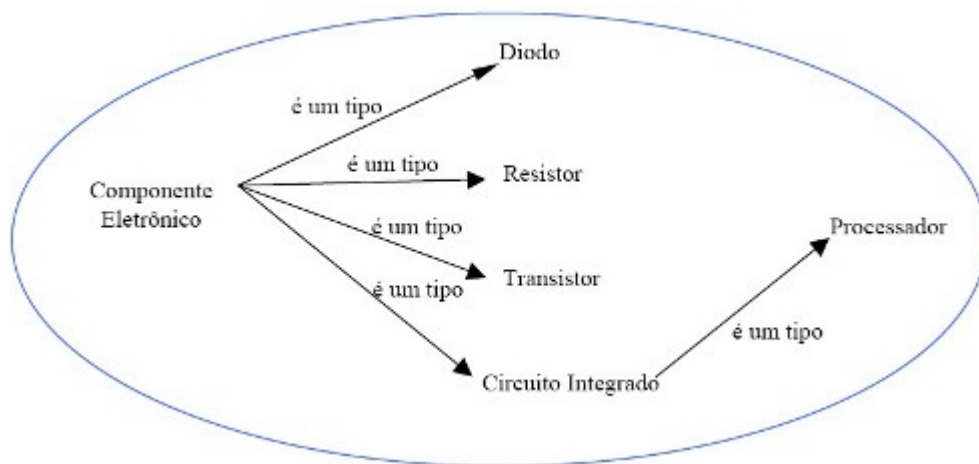


Figura 5.1: Exemplo de ontologia de domínio de circuitos eletrônicos

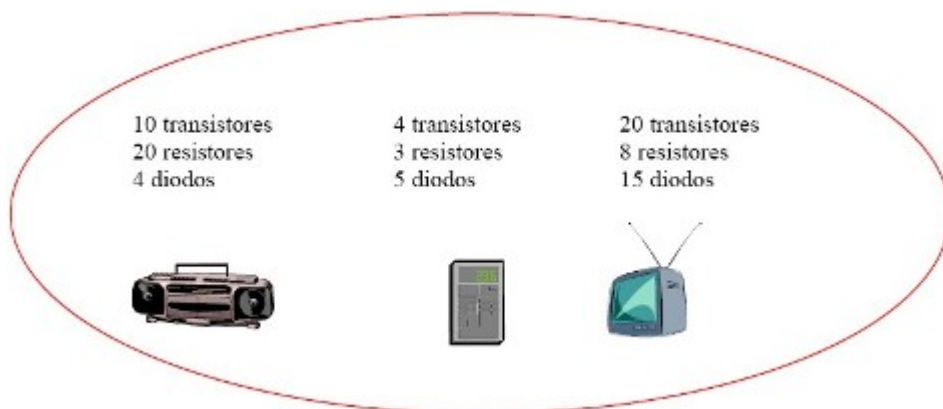


Figura 5.2: Exemplo da base de conhecimento que utiliza a ontologia de circuitos eletrônicos

5.2 Vantagens

Agora que já conhecemos algumas definições de ontologia, é importante que se destaque as vantagens de seu uso. Uma lista de vantagens da utilização de ontologias na ciência da computação é apresentada a seguir:

- Ontologias fornecem um vocabulário para representação do conhecimento. Esse vocabulário tem por trás uma conceitualização que o sustenta, evitando assim interpretações ambíguas.
- Ontologias permitem o compartilhamento do conhecimento. Por exemplo, considere uma ontologia para o domínio de livraria. Se ela estiver disponibilizada, várias livrarias podem usar o vocabulário fornecido por essa ontologia para construir os seus catálogos sem a necessidade de refazer uma análise do domínio de livraria.
- Fornece uma descrição exata do conhecimento; cria uma conceitualização comum de palavras; diminui diferentes interpretações sobre algo.
- Uma mesma conceitualização pode ser expressa em várias línguas.
- Pode-se estender o uso de uma ontologia genérica de forma a que ela se adeque a um domínio específico. Por exemplo, se alguém precisa de uma ontologia sobre carro para construir uma aplicação e só encontra uma ontologia genérica de veículos, podendo utilizar essa ontologia estendendo-a para o domínio específico da aplicação, que no caso é de carro.

5.3 Componentes de um Ontologia

Uma descrição completa sobre os componentes de uma ontologia proposta por Gómez-Pérez (1999) devido ao seu grau de formalismo (Puc-Rio (2003) "apud" Gómez-Peréz (1999)).

- Um conjunto de conceitos e uma hierarquia entre esses conceitos (taxinomia). Os conceitos podem ser abstratos; exemplo: força, ou concretos; exemplo: carro, elementares; exemplo: elétron ou compostos; exemplo: átomo, reais ou fictício.

Um exemplo de taxinomia é o conceito "homem" ser um subconceito do conceito "pessoa".

- Um conjunto de relacionamentos entre esses conceitos. Exemplo: pessoa e carro é o relacionamento de **ser dono do carro**.
- Fornece uma descrição exata do conhecimento; cria uma conceitualização comum de palavras; diminui diferentes interpretações sobre algo.

- Um conjunto de funções. Uma função é um caso especial de relacionamento em que o conjunto de elementos tem uma relação única com um outro elemento. Exemplo: ser-país-biológicos, onde um conceito "homem" e um conceito "mulher" estão relacionados a um conceito "pessoa".
- Um conjunto de axiomas, ou seja, regras que são sempre verdade. Exemplo: toda a pessoa tem uma mãe; toda pessoa morre.
- Um conjunto de instâncias que são um conhecimento prévio existente na ontologia. A grande maioria das instâncias está na base do conhecimento.

5.4 Classificação das ontologias

Será apresentado um sistema de classificação que usa a conceitualização como o principal critério para a classificação. Este modelo foi criado por Maedeché e propõe 4 tipos de classificação de uma ontologia:

- **Ontologias de alto-nível:**

Descrevem conceitos muito gerais como espaço, tempo, evento, etc. Esses conceitos tipicamente são independentes de um problema particular ou domínio. Sendo assim, é bem razoável ter-se uma ontologia de alto-nível compartilhada por grandes comunidades de usuários.

- **Ontologias de domínio:**

Descrevem o vocabulário relacionado a um domínio genérico, através da especialização de conceitos introduzidos nas ontologias de alto-nível. Exemplo: ontologias de veículos, documentos, etc.

- **Ontologias de tarefas**

Descrevem um vocabulário relacionado a uma tarefa ou atividade genérica, através da especialização de conceitos introduzidos nas ontologias de alto-nível.

- **Ontologias de aplicação**

São as ontologias mais específicas por serem utilizadas dentro das aplicações. Esse tipo de ontologia especializa conceito tanto das ontologias de domínio, como também das de tarefas. Exemplo: uma aplicação que trabalhe com carros de luxo. Essa ontologia especializará o conceito da ontologia de veículos (que é uma ontologia de domínio).

As ontologias de alto-nível possuem maior capacidade de reúso, por definir conceitos genéricos, enquanto as ontologias de aplicação são as que possuem menor capacidade de reúso, por definir conceitos relativos a uma aplicação específica.

A seguir é apresentada uma figura que ilustra os diferentes tipos de ontologias e suas relações

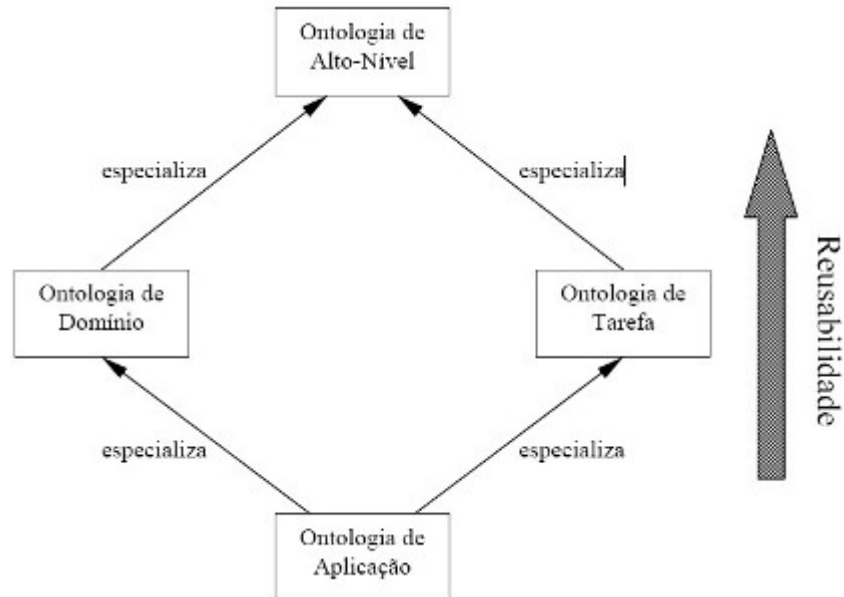


Figura 5.3: Diferentes tipos de ontologias e suas relações

5.5 Principais áreas de aplicação das ontologias

1. Processamento de linguagem natural:

O conhecimento do domínio é muito importante para compreensão do texto, que pode ser dado por meio de uma ontologia sobre o domínio de discurso do texto. O uso de ontologias é de vital importância por 2 motivos:

- Auxilia a elucidação de ambiguidade de compreensão existentes no texto. Com a utilização de uma ontologia sobre o domínio de discurso do texto se reduzem os problemas de ambiguidade.
- A ontologia funciona como um dicionário de conceitos dentro do domínio do texto.

2. Gestão de conhecimento:

Aquisição, manutenção e acesso ao conhecimento dentro de uma organização. A tecnologia de ontologias dentro dessa área auxilia das seguintes formas:

- Ontologias fornecem uma estrutura básica sobre a qual se constroem bases conhecimentos.

- Uma dificuldade dos sistemas de gestão de conhecimento é o fato de que muito do conhecimento presente dentro das organizações se encontra em uma forma não estruturada. Usando ontologias, pode-se notar informações semânticas em artefatos de informação não estruturados, visando assim a obtenção de resultados mais precisos em pesquisas de informação.

3. Web Semântica:

A tecnologia de ontologias entra para fornecer uma estrutura semântica para anotação das páginas da Web. Espera-se que uma estrutura fornecida pela Web semântica seja possível obter buscas mais precisas (uma vez que a semântica estará colocada em linguagem formal) e dar uma maior capacidade para os agentes de software que utilizem conteúdo da Web.

5.6 Abordagens para o desenvolvimento de uma ontologia

Assim como na atividade de desenvolvimento de software, o desenvolvimento de uma ontologia pode ser feito a partir de várias abordagens, cada uma apresentando suas vantagens e desvantagens.

Conforme o caso deve-se escolher a abordagem mais adequada, segundo a classe da ontologia a ser desenvolvida.

Algumas das principais abordagens para o desenvolvimento de uma ontologia, são apresentada a seguir:

1. **Abordagem partindo de uma inspiração:** O desenvolvedor começa com uma premissa sobre por que a ontologia é necessária. Usando a sua imaginação, criatividade e visão pessoal sobre o domínio de interesse da ontologia, o desenvolvedor constrói a ontologia que visa solucionar essa necessidade encontrada.

A principal vantagem dessa abordagem está no facto de ser possível obter uma ontologia inovadora para a resolução de um problema. A principal desvantagem dessa abordagem é que a ontologia gerada é resultado de uma visão muito subjetiva, do próprio desenvolvedor sobre o domínio, gerando assim dificuldades para a sua adoção.

2. **Abordagem indutiva:** A ontologia é desenvolvida através da observação, exame e análise de casos específicos no domínio de interesse da ontologia. A ontologia resultante para um caso específico é aplicada para todos os outros casos do mesmo domínio.

A principal vantagem dessa abordagem é ser possível utilizar uma mesma ontologia para solucionar uma série de casos dentro de um domínio. Porém a principal dificuldade é a incapacidade dessa ontologia em cobrir domínios menos específicos, ou seja, mais amplos.

3. **Abordagem dedutiva:** Nessa abordagem, a partir de uma ontologia mais genérica de um domínio, chega-se a uma ontologia mais específica e restritiva dentro de um subconjunto do domínio. A vantagem dessa abordagem está na capacidade de utilizar ontologias genéricas para geração de ontologias específicas. Porém a desvantagem é a necessidade de que já existe inicialmente uma ontologia genérica.
4. **Abordagem sintética:** O principal papel do desenvolvedor é fazer, de uma forma coerente, a composição de diversas ontologias em uma ontologia unificada. Por unir múltiplas ontologias, essa ontologia unificada tem grande propensão a ser aceita pelos usuários das ontologias usadas para a unificação. Dessa forma, essa ontologia unificada fornece uma forma coerente para a comunicação entre esses usuários das ontologias múltiplas. A dificuldade dessa abordagem está na dificuldade muitas vezes existente para estabelecer uma ligação de forma harmônica entre um conjunto grande de ontologias
5. **Abordagem colaborativa:** Nessa abordagem, o processo de desenvolvimento da ontologia não se encontra ao encargo de uma única pessoa e sim de um conjunto de pessoas. A vantagem está em aumentar as chances da ontologia se aceita como um padrão para um determinado domínio, pois possui a contribuição de um conjunto de pessoas para a sua formação. Outro ponto importante é o fato de a ontologia resultante desse processo ser muito elaborada do que nas abordagens supracitadas devido à mesma absorver o ponto de vista de mais de um indivíduo. A principal dificuldade está em coordenar o grupo de desenvolvimento, quando esse é muito grande.

5.7 Metodologias para desenvolvimento de uma ontologia

O processo de construção de ontologias ainda está pouco desenvolvido². Sendo que a maioria dos desenvolvedores usa os seus próprios critérios, e por isso muitos deles passam diretamente da fase de aquisição de conhecimento para a fase de implementação, o que pode causar os seguintes problemas:

- Os modelos conceituais ficam implícitos no código da implementação.
- Dificuldades de reuso da ontologia, porque as decisões de projeto estão implícitas no código.
- Problemas de comunicação devido às dificuldades que o expert no domínio da ontologia tem para entender o código da implementação.
- Dificuldades no desenvolvimento de ontologias complexas, pois a passagem da aquisição de conhecimento para a implementação é muito abrupta.

²Algumas ferramentas para construção de ontologias já têm sido desenvolvidas, como é o caso da ferramenta Protégé 2000 com Plug-In OWL (Ontology Web Language)

- Dependendo da linguagem escolhida para a codificação pode-se limitar a capacidade de descrição conceitual do domínio da ontologia.

Por isso é necessário que metodologias sejam adotadas para a criação de uma ontologia, afim de reduzirmos as dificuldades que foram citas acima.

Dessa forma, se faz necessária a adoção de uma metodologia afim de reduzir as dificuldades acima citadas. Uma metodologia será abordada na seção seguinte.

5.8 Metodologia para criação de uma ontologia

Esta metodologia foi proposta por Mariano Fernández, Asunción Gomez Pérez e Natalia Juristo. É uma metodologia bastante amadurecida, pois além de descrever mais a fundo a metodologia (passos seguidos e artefatos criados), fornece também um processo de desenvolvimento de ontologia e também propõe um ciclo de vida baseado em evolução de protótipos.

Como criar uma ontologia? Qual a metodologia criada por estes autores?

Eles criaram um conjunto de atividades que devem ser cumpridas e que auxiliam na construção de uma ontologia, este conjunto de atividades pode ser comparado ao conjunto de atividades que devem ser realizadas quando se pretende implementar um projeto de software.

- **Planejamento:** identifica as tarefas principais da ontologia, e planeja a utilização de recursos.
- **Especificação:** Define por que a ontologia está sendo construída, e quem serão os seus usuários.
- **Aquisição de conhecimento:** Adquire conhecimento sobre o domínio da ontologia. Pode ser feito: através de entrevistas com o expert na ontologia a ser criada, análise de livros sobre o domínio, etc.
- **Conceitualização:** criação de um modelo conceitual que descreve o problema e a sua solução.
- **Formalização:** transforma o modelo conceitual em um modelo formal.
- **Integração:** procurar integrar o máximo possível as ontologias existentes à nova ontologia.
- **Implementação:** implemente a ontologia em uma linguagem formal de modo que ela seja computável.
- **Avaliação:** Avalia a ontologia.
- **Documentação:** documentação da ontologia para facilitar futuro reuso e manutenção.

- **Manutenção:** Executar a manutenção da ontologia quando necessária.

As atividades de aquisição de conhecimento, avaliação e documentação, são executadas em todos os estágios do ciclo de vida. O planejamento é a primeira atividade a ser executada. A maior parte da atividade de aquisição é feita simultaneamente com o estágio de especificação da ontologia e vai decaindo conforme o ciclo de vida avança. A maior parte da atividade de avaliação da ontologia é feita durante os estágios iniciais do ciclo, de forma a diminuir a propagação de erro. A atividade de documentação deve ser realizada em todos os estágios.

É importante conhecer os artefactos gerados no fase de desenvolvimento e os que são gerados na parte de conceitualização da ontologia. Os artefatos gerados são de fácil compreensão para o desenvolvedor da ontologia e para o expert no domínio. Os artefatos também fornecem uma boa documentação sobre a ontologia.

O artefato gerado durante o estágio de especificação é um documento que cobre o objetivo principal, o propósito, a granularidade e o escopo da ontologia. Essa especificação deve ser completa e concisa.

Os principais artefatos propostos na metodologia que são gerados durante o estágio de especificação e conceitualização são:

- Glossário de termos: inclui todos os termos do domínio (conceitos, instâncias, atributos, etc) e suas descrições que devem ser claras e concisas.
- Árvore de classificação de conceitos: define relações tais como: subclasse e exclusividade mútua entre classes. Dessa forma são definidas várias taxinomias de domínio, sendo que cada uma deve gerar uma ontologia. Verificar a consistência (devem existir ciclos na árvore) e a concisão (não deve existir repetição de conceitos).
- Diagrama de relações binárias: esse diagrama estabelece os relacionamentos entre conceitos da mesma ontologia e de ontologias diferentes.
- Dicionário de conceito: documento que contém todos os conceitos do domínio, instâncias desses conceitos, e opcionalmente, sinónimos e antónimos dos conceitos, Para cada árvore de classificação deve existir um dicionário.
- Tabela de relações binárias: especifica o nome da relação, o nome dos conceitos de origem e destino da relação, a relação inversa e a cardinalidade da relação. Para cada árvore de classificação deve existir uma tabela desse tipo.
- Tabela de atributos de instância: descreve cada atributo de instância do dicionário de conceitos. Estes atributos são definidos nos conceitos e valorados nas instâncias. Para cada atributo de instância deve ser especificado: nome, tipo de valor, unidade de medida para valores numéricos, precisão de valores numéricos, faixa de valores aceites,

cardinalidade máxima e mínima. Atributos ou constantes utilizadas para inferir o valor do atributo, entre outras.

- Tabelas de atributos de classe: Descreve os atributos de classe no dicionário de conceitos. Esses atributos têm um valor que é dado pelo conceito, ou seja, o valor será o mesmo para todas as instâncias do conceito. Para cada atributo de classe deve ser especificado: nome do atributo, precisão dos valores numéricos, cardinalidade máxima e mínima, atributos que podem ser inferidos desse atributo e referências.
 - Tabela de axiomas: define conceitos por meio de expressões lógicas que são sempre verdadeiras.
 - Tabela de constantes: especifica para cada constante: nome, descrição em linguagem natural, o tipo lógico do valor, seu valor constante, sua unidade de medida, os atributos que podem ser inferidos usando a constante de referências.
 - Tabela de fórmulas: descreve cada fórmula das tabelas de atributos de instância. Essa tabela é usada para inferir os valores numéricos dos atributos de instância.
- Manutenção:** Executar a manutenção da ontologia quando necessária.
- Árvore de classificação de atributos: mostra graficamente os atributos e as constantes relacionados a uma sequência de cálculos de atributo raiz. É usado para validar que todos os atributos usados na fórmula fazem sentido e nenhum atributo foi omitido.
 - Tabela de instância: Descreve as instâncias do domínio.

Outras formas de se representar uma ontologia é fazendo uso de ferramentas para criação de ontologias, que apesar de poucas, elas podem ser facilmente obtidas através da internet e também podemos usar diagramas UML para representar ontologias.

5.9 Processos de aprendizagem baseados na interação social

Nesta secção pretende-se mostrar como será definido um modelo computacional para o processo de aprendizagem partindo de categorias de mediação e interação social através de uma ontologia. O desenvolvimento do sujeito seja este cognitivo, afetivo, social está estreitamente relacionado com a interação social. O foco de interesse aqui é o aspecto educacional e fator de influência que a interação social exerce sobre o desenvolvimento do sujeito (aluno).

Será mostrada uma ontologia criada para EaD baseada no perfil social do aluno, visto que este é foco do trabalho. Esta ontologia auxilia o aluno em suas ações de aprendizagem. E para isso é importante que se defina de um vocabulário e que ele seja conceitualizado. Para além da ontologia de aluno, também será criada uma ontologia para grupo de aluno, que possui regras que estabeleçam a comunicação entre os alunos pertencentes ao grupo.

A ontologia (aluno e grupo) proposta neste projeto tem por finalidade servir como base de conhecimento para elaboração de uma infra-estrutura de software de apoio a mediação da aprendizagem em ambientes de ensino a distância.

5.9.1 Desenvolvimento em uma perspectiva social

Para muitos autores como por exemplo Vygotsky (2001)(p.161 "apud" Severo et al. (2009) p.2) a ação do homem no mundo tem efeitos físicos de mudanças no mundo e efeitos psicológicos sobre o próprio homem. E wertsch("apud" Severo et al. (2009) p.2), diz que a mediação é um processo dinâmico, no qual intervém ferramentas e signos numa ação envolvendo o potencial das ferramentas para modelar a ação e os uso das mesmas por parte dos indivíduos.

Assim, do ponto de vista educacional , ações de mediação são importantes, não somente para o desenvolvimento cognitivo do aluno, mas também, para o desenvolvimento de sua autonomia. Em ambientes virtuais de ensino/aprendizagem não poderia ser diferente, eles apresentam mecanismos que permitem tanto ensino como aprendizagem flexível, independentes de espaço e tempo para que possam ser desenvolvidos Severo et al. (2009).

Além disso, aprendizagem, mediação e comunicação são conceitos intimamente ligados e que representam um ponto chave para a qualidade do ensino a distância. O professor exerce uma importante função no desempenho de sua atividade de mediação, buscando a realização de ajustes para que o aluno possa desenvolver um melhor desempenho, de acordo com as necessidades individuais do estudante.

O processo de regularização proposto por Diaz(1993 "apud" Severo et al. (2009) p.3) e adaptado por Passerino (2005 "apud" Severo et al. (2009) p.3) é realizado em níveis que partem de um controle, passando por autocontrole e chegando a auto-regulação. O controle é externo ao sujeito, realizado pelo sujeito mais experiente e pode assumir duas dimensões: direta ou indireta. O Controle Direto verifica-se através de ordens, diretivas e perguntas diretivas. Já o Indireto é feito através de perguntas perceptivas, conceituais, procedimentais, e culminam no afastamento físico que entra na categoria de auto-controle.

Segundo Diaz (1993), considera auto-controle, como sendo a realização, por parte do sujeito, de uma ação esperada obedecendo a um tutor internalizado. Ou seja, a figura do sujeito mais experiente que era real e externo no processo anterior, agora é interna. A auto-regulação não pode ser observada de forma direta , pois a mesma acontece internamente ao sujeito, mas considera-se que o sujeito está na categoria de auto-controle quando organiza, planeja e executa a ação sem intervenção de nenhum mediador externo. Assim a auto-regulação é o plano de ação concebido pelo sujeito que se converte no seu próprio tutor. A diferenciação central entre auto-controle e auto-regulação não passa pela internalização das ordens e diretivas do tutor, mas na capacidade emergente de planejar e definir objetivos próprios organizando funcionalmente sua conduta para os mesmos e adaptando-a de acordo com o contexto.

Algumas evidências de mediação em ambientes de ensino a distância serão apresentadas.

O trabalho de Hack (2009 "apud" Severo et al. (2009) p.3) aponta que as evidências de mediação em ambientes de ensino a distância mostram-se de grande importância no desenvolvimento de processos educativos. Os dados coletados a partir de pesquisa empírica mostraram que do ponto de vista pedagógico, o papel do mediador e seu ajuste nas ajudas oferecidas dentro das categorias identificadas de controle, auto-controle e auto-regulação são importantes para a autonomia do aluno e apropriação do conhecimento.

Segundo Severo, a hipótese básica do estudo é que as categorias de controle, auto-controle e auto-regulação, podem ser aplicadas de forma produtiva na análise e ajustes dos processos de mediação que ocorrem em ambientes de EaD, pois além de ajudarem na compreensão dos processos de mediação nesses ambientes, podem servir de base para a construção de ferramentas para ambientes de EaD que apoiem tanto professores quanto alunos a atingir processos de mediação produtivos, do ponto de vista pedagógico.

Consequentemente, num processo de educação, muitas vezes "os ajustes" na mediação são feitos no andamento dos trabalhos e seu acompanhamento e regulação só é possível a partir das "da leitura" do contexto intersubjetivo que se estabelece na interação social.

5.10 Ontologia para aluno e grupo de alunos

A seguir serão apresentadas ontologias para modelo de aluno e grupo de alunos que posteriormente serão modeladas como agentes (agentes inteligentes) através do JADE ³ que dará suporte a comunicação⁴ entre esses agentes (comunicação aluno-aluno e comunicação aluno-grupo ou vice-versa).

- **Ontologia de aluno:** define uma base de conhecimento para o aluno.

1. **Propriedades do aluno:** Aluno deve ter um id e um nome

2. **Ações do aluno:**

- Associar-se a um grupo (o grupo, para teste, será pré-existente);
- emitir material no grupo;
- responder ao grupo (como por exemplo, fórum);
- aluno pode responder diretamente a outro aluno;

³JADE é um framework do Java para o desenvolvimento de aplicações de agentes inteligentes

⁴A comunicação é estabelecida através dos protocolos de comunicação da FIPA

- **Ontologia do grupo:** define uma base de conhecimento acerca do grupo.

1. Propriedades do grupo:

- Deve ter um id e/ou nome e tipo de grupo, que define previamente a que tipo de matéria se destina o grupo. Por exemplo, podem ser criados um grupo de matemática, um de física, etc.
- Lista de alunos inseridos

2. Ações do grupo:

- Enviar mensagem de que o grupo foi criado com sucesso aquele que o criou;
- Enviar mensagem ao aluno que se associou a ele;
- Enviar mensagem de aviso quando há uma participação verbal(tipo fórum) no grupo;
- O grupo servirá de repositório de material;
- Emissão de material;
- Servirá para discussão de dúvidas entre os integrantes;
- O grupo deverá guarda o "Status"de participação dos integrantes (como cada participante está acessando material, enviando mensagens, etc);
- Estabelecer relatórios do tipo: quais os materiais (e frequência de acesso) acessados por cada integrante com quem o integrante se relaciona mais.

Como se pode perceber o desenvolvimento de uma ontologia inicia a execução de um projeto e a partir da sua definição é possível buscar a construção de um software para um ambiente virtual de ensino e aprendizagem, na tentativa de orientar o professor a estabelecer estratégias que visem melhorar a utilização do ambiente para a promoção da aprendizagem.

Capítulo 6

Agentes Inteligentes e a Educação

A Inteligência Artificial (IA) é uma área da Ciência da Computação que dá ênfase ao estudo de sistemas computacionais inteligentes, ou seja sistemas, que imitam o comportamento do ser humano - compreensão de linguagem, aprendizado, redes neurais, entre outras. Devido as aplicações computacionais da IA, muitos pesquisadores têm utilizado suas técnicas na área de educação, devido, principalmente, às suas potencialidades, no que se refere ao desenvolvimento de ambientes ensino-aprendizagem inteligentes(Bassani et al. (2007)).

Segundo Bassani et al. (2007), ambientes de ensino-aprendizagem inteligentes são sistemas que, na interação com o aluno, modificam suas bases de conhecimento através das percepções feitas. Possuem a capacidade de aprender e adaptar estratégias de ensino de acordo com o desempenho do aluno, e se caracterizam, sobretudo, por construir um Modelo Cognitivo desse aluno, através da interação, da formulação e da comprovação de hipóteses sobre seu conhecimento. Possui, contudo, a capacidade de adequar estratégias de ensino-aprendizagem ao aluno e à situação atual Viccari(1990) ("apud" Bassani et al. (2007) p.2). Sendo assim, estes ambientes computacionais têm um grande potencial para uso pedagógico, uma vez que envolve o usuário/aluno no processo, potencializando alto grau de interação. Desta forma podemos ver que o uso do computador na educação pode trazer inúmeras vantagens, tais como: estender o processo de ensino-aprendizagem; dar uma maior qualidade no processo de Educação à Distância (EaD) e ser uma boa ferramenta para o paradigma construcionista, entretanto é importante referir que este também traz desvantagens no que concerne a ausência de contato presencial dificultando o acompanhamento e avaliações informais, porém, podemos usar o próprio sistema (inteligente) que vai auxiliar o professor na tarefa de acompanhar e avaliar o aluno.

A seguir é apresentada uma ilustração do processo Ensino-Aprendizagem Auxiliado pelo Computador.

Um dos ramos da IA é a Inteligência Artificial Distribuída (IAD) que soluciona problemas partindo da individualidade para a coletividade, ou seja, a inteligência é vista como emergente da ação e interação de entidades (agentes) autônomas.

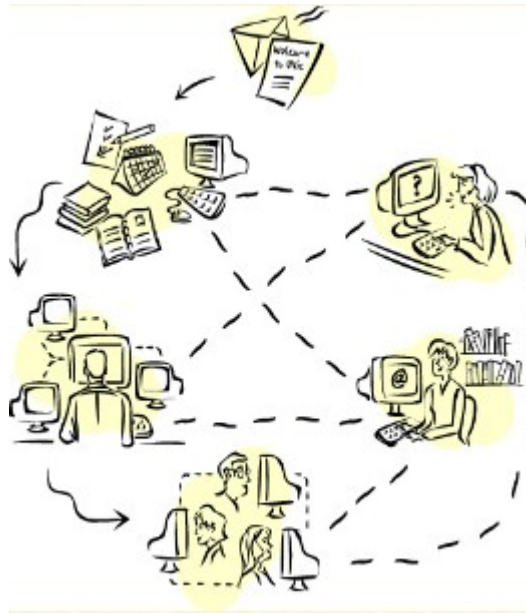


Figura 6.1: Uso do computador na EaD

Segundo alguns autores, a IAD está dividida em 3 áreas: **Resolução Distribuída de Problemas (RDP)**; **Inteligência Artificial Paralela (IAP)** e **Sistemas Multiagentes (SMA)** que é o foco do nosso trabalho, portanto, os itens seguintes abordarão os seus conceitos básicos, mas antes vale ressaltar o que significa Sistemas Multiagentes.

Os Sistemas Multiagentes são caracterizados pela existência de um certo número de entidades autônomas (agentes), heterogêneas e potencialmente independentes, trabalhando juntas para resolver um problema.

6.1 Agentes Inteligentes

Definir agentes com um conceito único ou padrão é muito difícil, pois existem várias abordagens e ponto de vista de diferentes autores. Além disso, devido às suas mais diversas aplicações uma definição precisa torna-se cada vez mais complicada e variada.

RUSSEL e NORVIG (1995) definem um agente como um sistema capaz de perceber, através de sensores, e agir, através de atuadores, em um dado ambiente. Um agente também pode ser uma entidade à qual se atribuem estados, denominados estados mentais, tais como crenças, decisões, capacidades, compromissos e objetivos (conceitos análogos ou similares aos humanos) (SHOHAM (1993)).

Uma definição bem completa é dada por Bradshaw (1997), que descreve um agente como sendo uma entidade de software que funciona de forma contínua e autônoma em um ambiente. Segundo este autor, um agente também deve ser capaz de perceber e atuar no seu ambiente, de forma flexível e inteligente, sem requerer intervenção ou orientação humana constantes.

Idealmente, um agente deve aprender através da experiência, comunicar-se e cooperar com outros agentes que porventura coexistam no mesmo ambiente. Ainda, um agente deve poder mover-se de um local para outro a fim de satisfazer os seus objetivos.

Um sistema pode utilizar-se da interação entre vários agentes (modelo multiagente).

A partir disto, podemos destacar as seguintes propriedades para agentes:

- **Autonomia:** Um agente deve agir sem intervenção humana direta, portanto deve possuir algum tipo de controle sobre suas ações e seu estado interno;
- **Reatividade:** Um agente deve ser capaz de reagir aos estímulos externos produzidos pelo seu ambiente ou por outros agentes;
- **Pró-atividade:** Um agente não somente reage ao seu ambiente, mas também deve exibir um comportamento orientado à satisfação de seus objetivos (Orientação a Objetivos).
- **Intencionalidade:** Capacidade de representação explícita dos seus objetivos;
- **Racionalidade:** Habilidade de agir de forma a atingir seus objetivos e não contra eles;
- **Continuidade temporal:** Persistência de identidade por longos períodos de tempo;
- **Sociabilidade:** Habilidade de interação com outros agentes através de mecanismos de comunicação;
- **Benevolência:** Capacidade de cooperação com outros agentes;
- **Adaptabilidade:** Capacidade de aprender através da experiência;
- **Mobilidade:** Capacidade de mover-se de um ambiente para outro.

Em determinadas aplicações, algumas características são mais importantes que as outras, portanto, um agente não precisa ter necessariamente todas elas. O comportamento do agente é dado pelo ambiente no qual ele está inserido e modelado através do seu respectivo projeto.

A FIGURA 7.2 ilustra de forma esquemática a noção elementar de um agente. Nesta figura podemos ver alguns dos atributos citados no parágrafo anterior como reatividade (percepções e ações) e pró-atividade (estado interno e ações).

6.2 Categoria de Agentes

Para melhor compreensão da aplicabilidade da tecnologia de agentes, é interessante referenciar sobre os diferentes tipos de agentes e suas diferenças, a fim de termos uma noção mais clara da utilidade no emprego de agentes. A classificação dos agentes pode ser feita quanto à mobilidade, ao relacionamento entre agentes e quanto a capacidade de raciocínio.

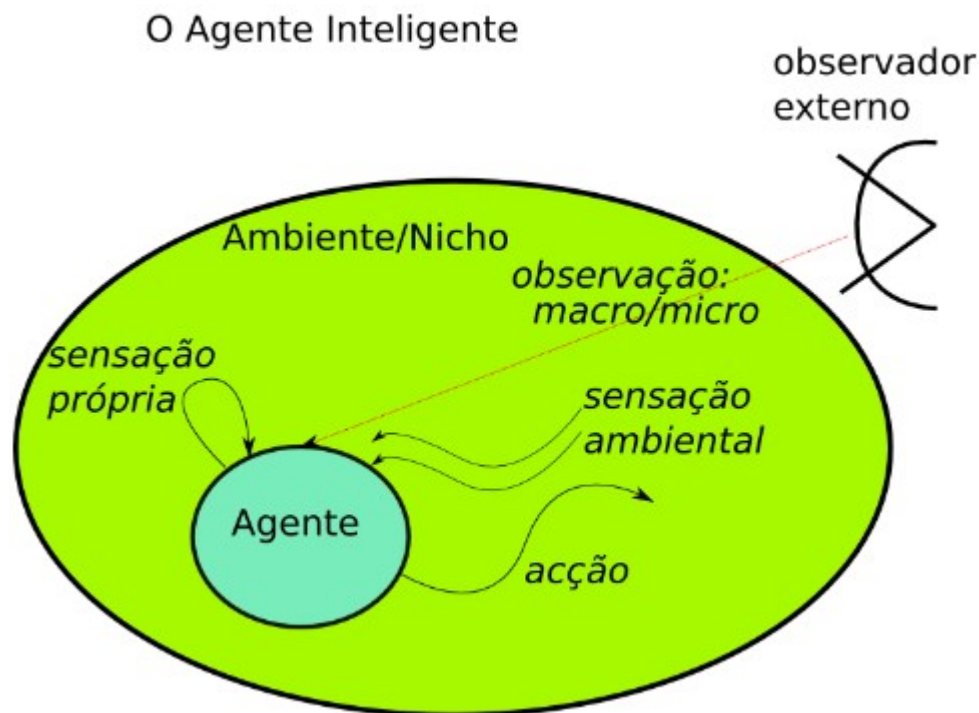


Figura 6.2: Esquema de agente

- **Agentes Móveis:** são agentes que tem a mobilidade como característica principal. Isto é, uma capacidade de mover-se seja por uma rede interna local (intranet) ou até mesmo pelo Web, transportando-se pelas plataformas levando dados e códigos. Seu uso tem crescido devido alguns fatos como uma heterogeneidade cada vez maior das redes e seu grande auxílio em tomadas de decisões baseadas em grandes quantidades de informação.
- **Agentes situados ou estacionários:** são aqueles opostos aos móveis. Isto é, são fixo em um mesmo ambiente e ou plataforma. Não se movimentam em uma rede e muito menos na Web.
- **Agentes Competitivos:** são agentes que "competem" entre si para a realização de seus objetivos ou tarefas. Ou seja, não há colaboração entre os agentes.
- **Agentes Coordenados ou Colaborativos:** agentes com a finalidade de alcançar um objetivo maior, realizam tarefas específicas porém coordenando-as entre si de forma que suas atividades se completem.
- **Agentes Reativos:** é um agente que reage a estímulos sem ter memória do que já foi realizado no passado e nem previsão da ação a ser tomada no futuro. Não tem representação do seu ambiente ou de outros agentes e são incapazes de prever e antecipar ações. Geralmente atuam em sociedades como uma colônia de formiga por exemplo.

Baseiam-se muito também na "teoria do caos" no qual afirma que até mesmo no caos existe uma "certa organização". No caso da formiga, por exemplo, uma única dela não apresenta muita inteligência mas quando age no grupo comporta-se o todo como uma entidade com uma certa inteligência. Ou seja, a força de um agente reativo vem da capacidade de formar um grupo e construir colônias capazes de adaptar-se a um ambiente.

- **Agentes Cognitivos:** esses, ao contrario dos agentes reativos, podem raciocinar sobre as ações tomadas no passado e planejar ações a serem tomadas no futuro. Ou seja, um agente cognitivo é capaz de "resolver" problemas por ele mesmo. Ele tem objetivos e planos explícitos os quais permitem atingir seu objetivo final. Ferber afirma que para que isso se concretize, cada agente deve ter uma base de conhecimento disponível, que compreende todo os dados e todo o "know-how" para realizar suas tarefas e interagir com outros agentes e com o próprio ambiente. Sua representação interna e seus mecanismos de inferência o permitem atuar independentemente dos outros agentes e lhe dão uma grande flexibilidade na forma de expressão de seu comportamento. Além disso, devido a sua capacidade de raciocínio baseado nas representações do mundo, são capazes de ao mesmo tempo memorizar situações, analisálas e prever possíveis reações para suas ações.

6.3 Aplicações dos Agentes

Algumas das aplicações dos agentes inteligentes serão apresentadas de seguida:

- **Agentes no processo ensino-aprendizagem:** Utilização de agentes no aprendizado e no processo de treinamento de usuários.
- **Agentes na indústria:** A grande vantagem da aplicação de agentes na produção industrial é o fato de que muitos dos sistemas que atuam neste propósito são complexos e isolados. O papel dos agentes seria integrá-los, afim de compartilhar informações entre os sistemas.
- **Agentes em simulação:** Agentes podem simular situações dando um grau maior de veracidade. Tanto na área de entretenimento quanto na área de pesquisa tecnológica e militar.
- **Agentes em realidade virtual:** Nessas aplicações o agente atua como um participante que auxilia e monitora certas atividades e usuários, assistindo-os ou ajudando-os quando necessário, principalmente em ambientes virtuais muito complexos.

- **Agentes na prestação de serviços:** Nesse contexto os agentes irão agregar valor a certos serviços, gerenciando a informação a fim de satisfazer as necessidades dos clientes. Podemos citar por exemplo:
 1. manutenção e atualização de bases de dados explorando a informação (data mining);
 2. comércio eletrônico, onde agentes procuram preços mais convenientes ou ofertas e produtos com as especificações desejadas pelo usuário;
 3. gerência de correio eletrônico e filtragem de mensagens, funcionando como um assistente pessoal;
 4. gerência de produção e muitas outras aplicações;
- **Agentes em redes de computadores:** As suas funções nessas aplicações variam muito. Varia desde definição de melhores rotas, controle de manutenção de equipamentos, gerenciamento de dados até a monitoramento de atividades e gerência da própria rede.

6.4 Sistemas Multiagentes

Um único agente pode requerer muito conhecimento para resolver problemas complexos. Em alguns casos, o problema é tão complexo que um agente não pode, por ele mesmo, resolvê-lo (BRENNER et al. (1998)). Além disso, muitos problemas têm características distribuídas e, portanto, necessitam de unidades distribuídas de conhecimento para a sua resolução. Portanto, seria interessante a existência de unidades independentes que resolvessem cada questão individualmente e que juntos resolvessem o problema todo, ou seja, um sistema composto por mais de um agente que trabalham juntos para alcançarem um objetivo comum. Esses sistemas são denominados Sistemas Multiagentes (SMA).

Em um SMA, cada agente tem os seus próprios objetivos e contata outros agentes para obter informações ou contribuir na solução de um problema maior. Logo, sob o ponto de vista de constituição de sociedade de agentes, a fundamentação dos SMA é baseada na interação social de seus indivíduos.

Segundo BRENNER et al. (1998), uma vantagem dos sistemas multiagentes, é o fato deles permitirem a integração de agentes já existentes. Isso faz com que a solução do problema não demande o desenvolvimento de novos e especializados agentes, ou seja, o conhecimento dos agentes existentes pode ser utilizado para resolver determinado problema.

O conhecimento especializado de cada agente não pode ser utilizado por outro agente e nenhuma estratégia comum de solução pode ser desenvolvida sem mecanismos de comunicação e cooperação (BRENNER et al. (1998)). Deste modo, a resolução de problemas através de SMA só é possível quando os agentes têm capacidades de se comunicarem e de cooperarem

uns com os outros. Além disso, é necessária alguma forma de coordenação do comportamento dos agentes.

6.5 Comunicação entre agentes

A comunicação é fundamental para permitir que haja colaboração, negociação, cooperação entre agentes.

Em sistemas multiagentes, é necessário que a comunicação seja disciplinada para que os objetivos sejam alcançados efetiva e eficientemente, necessitando assim uma linguagem que possa ser entendida pelos outros agentes presentes no ambiente. Essa comunicação tem como principal objetivo a partilha do conhecimento com os outros agentes e a coordenação de atividades entre eles, ou seja, ela deve permitir que agentes troquem informações entre si e coordenem suas próprias atividades resultando sempre em um sistema coerente.

Existem diversas maneiras para agentes trocar informações uns com os outros em sistemas multiagentes. Para Baker (1997) agentes podem trocar mensagens diretamente, chamada também por alguns autores como comunicação direta, podem comunicar-se através de um agente "facilitador" especial em sistema "federado" (comunicação assistida), podem também utilizar uma comunicação por difusão de mensagens (broadcast) e até utilizar o modelo de comunicação através de blackboard ou quadro negro.

6.5.1 Comunicação direta

A comunicação direta, ou comunicação via troca de mensagens direta, cada agente comunica diretamente com qualquer outro agente sem qualquer intermediário. Na Figura 8.1 há um estabelecimento de uma ligação direta (ponto-a-ponto) entre os agentes através de um conjunto de protocolos que garante a chegada de mensagens com segurança. Nesse tipo de comunicação faz-se necessário que cada agente envolvido tenha conhecimento da existência dos outros agentes e da forma de como endereçar mensagens para eles. A principal vantagem deste tipo de comunicação entre agentes é o fato de não existir um agente coordenador da comunicação. Isso porque esses agentes coordenadores podem levar a um "gargalo" ou até ao bloqueio do sistema caso haja, por exemplo, um grande número de troca de mensagens. Entretanto, algumas desvantagens podem ser verificadas como o custo da comunicação se tornar muito grande, principalmente quando há um grande número de agentes no sistema, e a própria implementação que se torna muito complexa em comparação às outras formas de comunicação.

6.5.2 Comunicação por sistemas federados

Na comunicação por sistemas federados Baker (1997) ou comunicação assistida, os agentes utilizam algum sistema ou agente especial para coordenar suas atividades. Ou seja, uma estrutura hierárquica de agentes é definida e a troca de mensagens dá-se através agentes

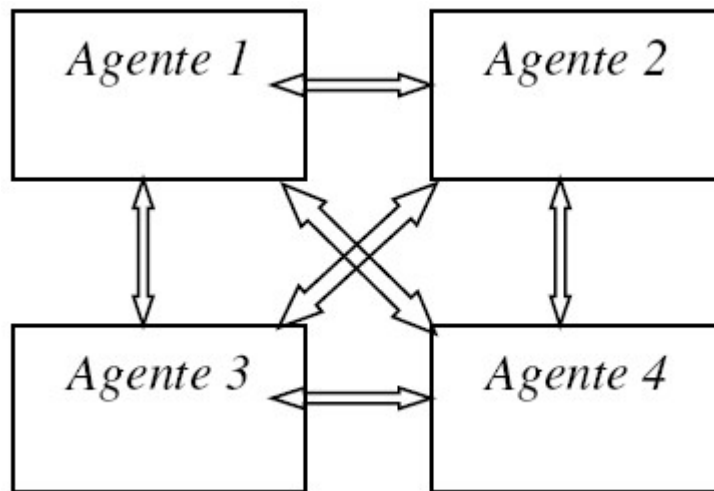


Figura 6.3: Comunicação Direta entre agentes

especiais designados "facilitadores" ou mediadores (Veja a figura 8.2). Essa é uma alternativa bem popular à comunicação direta pois diminui muito o custo e a complexidade necessária aos agentes individuais na realização da comunicação. Geralmente é utilizado quando o número de agentes dentro do sistema é muito grande.

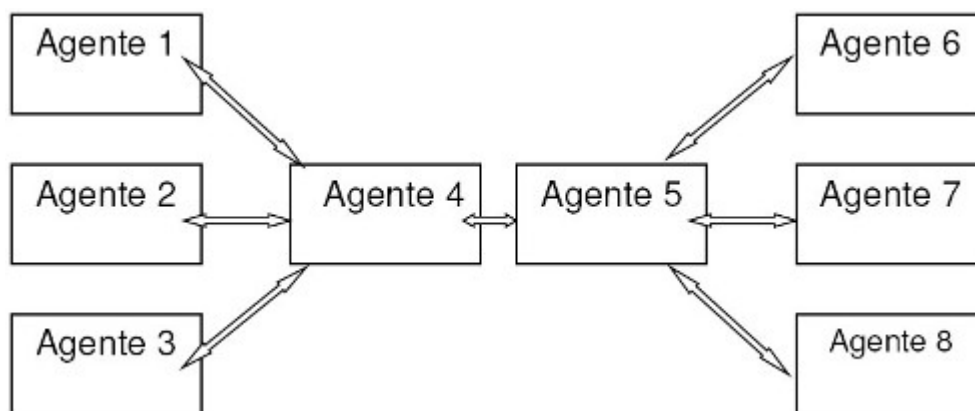


Figura 6.4: Comunicação por sistema federado ou comunicação assistida

6.5.3 Comunicação por broadcast

A comunicação por difusão de mensagens ou broadcast geralmente é utilizada em situações onde a mensagem deve ser enviada para todos os agentes do ambiente ou quando o agente remetente não conhece o agente destinatário ou seu endereço. Em suma, todos os agentes recebem a mensagem enviada.

6.5.4 Comunicação por blackboard

Comunicação por quadro negro ou blackboard ou ainda quadro de avisos, segundo Baker (1997), é bastante usada na Inteligência Artificial como modelo de memória compartilhada. Ou seja, nada mais é que um repositório onde os agentes escrevem mensagens a outros agentes e obtêm informações sobre o ambiente.

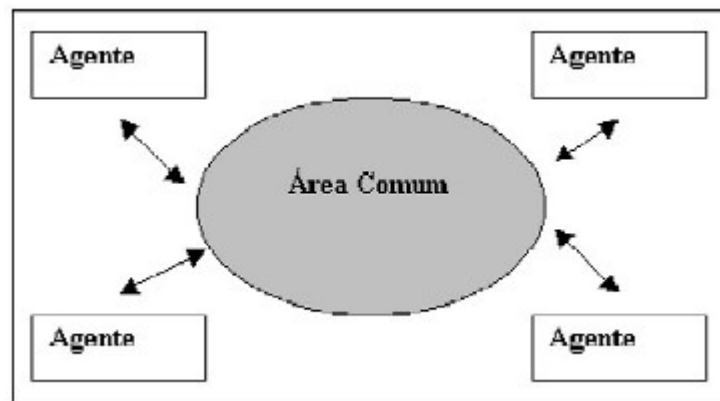


Figura 6.5: Comunicação por quadro negro

6.6 Protocolos de comunicação entre agentes

O mecanismo de troca de mensagens necessita da definição de um protocolo de comunicação que especifique o exato processo de comunicação, um formato para as mensagens e uma linguagem de comunicação. Esses requisitos nos levam às Linguagens de Comunicação de Agentes (ACL) e aos serviços de transportes de mensagens.

Uma ACL determina um meio através do qual uma atitude (afirmação, requisição, consulta, etc) a respeito do conteúdo da troca de conhecimentos é comunicada (Finin et al. (1997)).

Finin et al. (1997) destaca que para uma linguagem de comunicação possa ser adequada à comunicação entre agentes, ela deve possuir as seguintes propriedades quanto a:

- **Agentes forma:** Uma ACL deve ser declarativa, sintaticamente simples e legível. Ela deve proporcionar fácil realização de *parsing* ;
- **O conteúdo:** Uma ACL deve fornecer um conjunto bem definido de ações de comunicação (primitivas). Além disso, ela deve poder ser estendida de forma a se adaptar bem com outros sistemas;
- **A semântica:** Uma ACL não deve ter uma semântica ambígua, ela deve ser baseada numa teoria e considerar tempo e local. A descrição da semântica é normalmente feita através da linguagem natural;

- **A implementação:** Uma ACL deve ser eficiente tanto na velocidade como na boa utilização da largura de banda da rede. Deve possuir uma interface amigável, isto é, detalhes devem ficar escondidos. Deve também ser possível realizar-se uma implementação parcial da linguagem para que agentes possam utilizar apenas um subconjunto de primitivas de ações de comunicação;
- **A rede:** Uma ACL deve adaptar-se bem com as tecnologias atuais de rede. Deve suportar os tipos básicos de conexão ponto-a-ponto, multicast, broadcast. Conexões síncronas e assíncronas também devem ser suportadas. Um conjunto rico de primitivas deve ser disponibilizado de forma que possam ser desenvolvidos linguagens e protocolos de alto-nível e esses mecanismos devem ser independentes da camada de transporte;
- **O ambiente:** Os ambientes em que os agentes inteligentes são utilizados freqüentemente são distribuídos, heterogêneos e dinâmicos. Portanto, deve ser possível a integração com outras linguagens e protocolos;
- **A confiabilidade:** A linguagem deve proporcionar uma comunicação confiável e segura entre os agentes.

O KQML (*Knowledge Query and Manipulation Language*) é um protocolo para troca de informações e conhecimento entre agentes. Na prática, até o ano de 2000, quando foi lançada a primeira implementação da FIPA-ACL, ela era a única linguagem padronizada e com implementação disponível e, portanto, dominava a área de comunicação entre agentes.

A FIPA-ACL é uma linguagem de comunicação de agentes que vem ganhando espaço. Ela foi apresentada inicialmente como uma alternativa bem fundamentada para a KQML. Porém, apesar de ter sido proposta em 1997, foi somente a partir do ano 2000, com os lançamentos do padrão FIPA 2000 e da plataforma FIPA, que ela passou a ter uma implementação disponível e a competir com a KQML.

Na próxima seção será feita uma maior abordagem sobre a plataforma FIPA, uma vez que o JADE, ferramenta a ser utilizada para a implementação de aplicação de agentes, proposta neste trabalho.

Capítulo 7

FIPA

A linguagem FIPA (Foundation for Intelligent Physical Agents) foi apresentada em 1996 em Genebra, Suíça com o propósito de desenvolver e padronizar as tecnologias de sistemas multiagentes promovida pela organização FIPA (FIPA (2010)). A linguagem FIPA é open source e a estrutura de suas mensagens é muito parecida com a estrutura de mensagens da KQML.

Cada mensagem é composta por um ato comunicativo (performative) e um conjunto de parâmetros, dos quais o único obrigatório é a performativa. Porém, a grande maioria das mensagens contém um emissor, um receptor e um campo de conteúdo. Os parâmetros podem ser alocados em qualquer posição dentro da mensagem.

7.1 Parâmetros das mensagens do FIPA-ACL

Os parâmetros nativos da FIPA-ACL são divididos em cinco categorias: tipo do ato comunicativo, participantes da comunicação, conteúdo de mensagem, descrição do conteúdo e controle de conversação.

A lista completa dos parâmetros e suas respectivas categorias são mostradas na TABELA 8.1:

O conteúdo de uma mensagem, indicado pelo parâmetro content, referencia a informação sobre a qual o ato comunicativo se aplica. Em geral, o conteúdo pode ser expresso em qualquer linguagem. A linguagem utilizada na representação do conteúdo pode ser declarada no parâmetro language. O projeto FIPA define e sugere algumas linguagens de representação de conteúdo padrões, tal como a linguagem SL.

7.2 Especificações FIPA

As especificações produzidas pela FIPA são publicadas em documentos que seguem uma identificação especial quanto ao seu tipo e lançamento. Nesta seção serão apresentados esses documentos. As especificações são divididas em cinco áreas distintas: *Aplicações de Sistemas*

Multiagentes, Arquiteturas Abstratas, Comunicação entre Agentes, Gerenciamento de Sistemas Multiagentes e Transporte de Mensagens entre Agentes. A seguir será feita uma análise de cada uma dessas categorias.

7.2.1 Aplicações de Sistemas Multiagentes

As Aplicações de Sistemas Multiagentes são exemplos de domínios onde podem ser utilizados agentes FIPA. Possui definições de ontologias e descrições de serviços para esses domínios.

O padrão FIPA-97 já incluía especificações de aplicações para PTA (Personal Travel Assistance) entretenimento audiovisual, gerenciamento de redes e assistência pessoal. No padrão FIPA-2000 essas especificações foram mantidas, tendo sido incluídas apenas duas novas: suporte a aplicações nômades (agentes móveis ou migratórios) e bufferização de mensagens. Logo após, foi adicionada a especificação de Qualidade de Serviço. A tabela abaixo mostra esses documentos.

7.2.2 Arquitetura Abstrata para Sistemas Multiagentes

A Arquitetura abstrata para Sistemas Multiagentes trabalha com entidades abstratas necessárias para a construção de serviços e de um ambiente de agentes. Assim, ela define quais são as características arquiteturais que um sistema multiagente deve estar conforme especificando os requisitos para arquiteturas concretas (implementadas em JAVA, PROLOG, etc.) de sistemas multiagentes.

É incluída, também, uma especificação definindo como sistemas multiagentes poderiam ser estruturados em domínios distintos e como poderiam ser estabelecidas políticas de manutenção.

7.2.3 Comunicação entre Agentes

O maior avanço e detalhamento em termos de padronização ocorreram no tratamento das questões relativas à comunicação entre agentes. Entre a especificação da linguagem FIPA-ACL feita por um único documento (considerado atualmente obsoleto, a especificação (FIPA00003, 1997)[FIP 2003]) em 1997 e sua instanciação atual, coberta por quase trinta documentos distintos, percebe-se claramente onde esteve concentrado a maior parte do trabalho da FIPA nesse intervalo de tempo. Mesmo a arquitetura abstrata também poderia ser vista, em grande parte, como um trabalho necessário para uma melhor fundamentação (pragmática) dos processos de comunicação.

O formato básico que deve ser seguido por todas as mensagens FIPA-ACL foi descrito em (FIPA00061, 2001) FIPA (2010).

Esta categoria divide-se em:

- **Atos comunicativos:** define os atos comunicativos padrão da linguagem FIPA-ACL;

Tipo do ato comunicativo	
<i>Performative</i>	Denota o tipo do ato comunicativo da mensagem.
Participantes da Comunicação	
<i>Sender</i>	É o emissor da mensagem.
<i>Receiver</i>	É o receptor da mensagem.
<i>Reply-to</i>	Indica que as próximas mensagens dessa conversação deverão ser enviadas para o agente indicado pelo parâmetro reply-to.
Conteúdo de mensagem	
<i>Content</i>	É o conteúdo ou conhecimento transportado pela mensagem.
Descrição do conteúdo	
<i>Language</i>	É a linguagem no qual o conhecimento está expresso.
<i>Encoding</i>	Aponta a codificação utilizada na expressão da linguagem de conteúdo.
<i>Ontology</i>	É a ontologia utilizada para dar significado à expressão de conteúdo.
Controle de conversação	
<i>Protocol</i>	É o protocolo de interação utilizado para essa mensagem pelo agente emissor.
<i>Conversation-id</i>	Introduz uma expressão que será usada para identificar a conversação em andamento.
<i>Reply-with</i>	Introduz uma expressão que será usada pelo agente que responderá a essa mensagem para identificá-la.
<i>In-reply-to</i>	É a expressão que referencia a mensagem à qual se está respondendo.
<i>Reply-by</i>	Explicita um tempo máximo durante o qual o agente emissor estará esperando por uma resposta a esta mensagem.

Tabela 7.1: Parâmetros das mensagens FIPA-ACL

Identificador	Título
SI00014	FIPA Nomadic Application Support Specification
XC00079	FIPA Agent Software Integration Specification
XI00080	FIPA Personal Travel Assistance Specification
XI00081	FIPA Audio-Visual Entertainment and Broadcasting Specification
XI00082	FIPA Network Management and Provisioning Specification
XI00083	FIPA Personal Assistant Specification
XC00092	FIPA Message Buffering Service Specification
SC00094	FIPA Quality of Service Specification

Tabela 7.2: Especificações FIPA: Aplicações de Sistemas Multiagentes

Identificador	Título
SC00001	FIPA Abstract Architecture Specification
PC00089	FIPA Domains and Policies Specification

Tabela 7.3: Especificações FIPA: Arquitetura Abstrata

Identificador	Título
SC00023	FIPA Agent Management Specification

Tabela 7.4: Especificações FIPA: Gerenciamento de agentes

- **Protocolos de Interação:** Define uma seqüência lógica de troca de mensagens, especificando atos comunicativos para cada uma delas. Os protocolos de interação são úteis quando alguma conversação necessita um maior poder de interação entre os agentes.
- **Linguagem de Conteúdo:** define as diferentes maneiras de representar ou codificar a informação passada através de uma mensagem ACL;

7.2.4 Gerenciamento de Agentes

A categoria de Gerenciamento de Agentes trabalha com a especificação de ferramentas para controle e gerenciamento de agentes em plataformas de agentes.

A plataforma de Gerenciamento de Agentes descrita no documento mostrado pela TABELA abaixo que define o ambiente onde os agentes FIPA existem e operam. Ele estabelece o modelo lógico de referência para a criação, registro, localização, comunicação, migração e desativação dos agentes.

7.2.5 Transporte de Mensagens entre Agentes

O Transporte de Mensagens entre Agentes se preocupa em especificar a forma como as mensagens são transportadas e representadas através de diferentes protocolos de rede.

Esta categoria está dividida em:

- **Representações da ACL:** define diferentes formas de representar uma mensagem ACL;
- **Representações de Envelope:** define diferentes formas de representar um envelope;
- **Protocolos de Transporte:** define formas de transporte de mensagens ACL para diferentes protocolos de redes.

O FIPA-OS é um ambiente que está tendo um crescimento muito grande, com a sua utilização em laboratórios de pesquisa e instituições de ensino no mundo todo. Isso pode ser observado na agilidade, na divulgação e correção de erros, e na disponibilização de material para testes e compreensão do ambiente.

Transporte de Mensagens entre Agentes	
SC00067	FIPA Agent Message Transport Service Specification
XC00093	FIPA Messaging Interoperability Service Specification
Representações da ACL	
SC00069	FIPA ACL Message Representation in Bit-Efficient Specification
SC00070	FIPA ACL Message Representation in String Specification
SC00071	FIPA ACL Message Representation in XML Specification
Representações de Envelope	
SC00085	FIPA Agent Message Transport Envelope Representation in XML Specification
SC00088	FIPA Agent Message Transport Envelope Representation in Bit Efficient Specification
Protocolos de Transporte	
SC00075	FIPA Agent Message Transport Protocol for IIOP Specification
XC00076	FIPA Agent Message Transport Protocol for WAP Specification
SC00075	FIPA Agent Message Transport Protocol for HTTP Specification

Tabela 7.5: Especificações FIPA: Transporte de Mensagens entre Agentes

Capítulo 8

JADE

Jade ¹ (*Java Agent DEvelopment framework*) é um ambiente para desenvolvimento de aplicações baseada em agentes conforme as especificações da FIPA (*Foundation for Intelligent Physical Agents*) para interoperabilidade entre sistemas multiagentes totalmente implementado em Java. Foi desenvolvido e suportado pelo CSELT da Universidade de Parma na Itália. É open source (LGPL ²) Segundo Telecom-Itália (2010), o principal objetivo do Jade é simplificar e facilitar o desenvolvimento de sistemas multiagentes garantindo um padrão de interoperabilidade entre sistemas multiagentes através de um abrangente conjunto de agentes de serviços de sistema, os quais tanto facilitam como possibilitam a comunicação entre agentes, de acordo com as especificações da FIPA: serviço de nomes (naming service) e páginas amarelas (yellow-page service), transporte de mensagens, serviços de codificação e decodificação de mensagens e uma biblioteca de protocolos de interação (padrão FIPA) pronta para ser usada. Toda sua comunicação entre agentes é feita via troca de mensagens. Além disso, lida com todos os aspectos que não fazem parte do agente em si e que são independentes das aplicações tais como transporte de mensagens, codificação e interpretação de mensagens e ciclo de vida dos agentes. Ele pode ser considerado como um "middle-ware" de agentes que implementa um framework de desenvolvimento e uma plataforma de agentes. Em outras palavras, uma plataforma de agentes em complacência com a FIPA e um pacote, leia-se bibliotecas, para desenvolvimento de agentes em Java.

¹Jade é uma marca registrada do TILAB (<http://www.telecomitalialab.com>). Foi desenvolvido pelo TILAB juntamente com o AOT (<http://aot.ce.unipr.it>) com a permissão do TILAB.) anteriormente conhecido com CSELT.

²LGPL ou Lesser General Public License.

8.1 Características do JADE

Seguem abaixo algumas características do framework JADE:

- *Plataforma distribuída de agentes:* JADE pode ser dividida em vários "hosts" ou máquinas (desde que eles possam ser conectados via RMI). Apenas uma aplicação Java e uma Java Virtual Machine é executada em cada host. Os agentes são implementados como threads Java e inseridos dentro de repositórios de agentes chamados de containeres (Agent Containers) que provêm todo o suporte para a execução do agente.
- *Gui ou Graphical User Interface:* Interface visual que gerencia vários agentes e containeres de agentes inclusive remotamente.
- *Ferramentas de Debugging:* ferramentas que ajudam o desenvolvimento e depuração de aplicações multiagentes baseados em JADE. Suporte a execução de múltiplas, paralelas e concorrentes atividades de agentes - através dos modelos de comportamentos (Behaviours).
- *Ambiente de agentes complacente a FIPA:* No qual incluem o sistema gerenciador de agentes (AMS - Agent Management System), o diretório facilitador (DF - Directory Facilitator) e o canal de comunicação dos agentes (ACC - Agent Communication Channel). Todos esses três componentes são automaticamente carregados quando o ambiente é iniciado.
- *Transporte de mensagens:* Transporte de mensagens no formato FIPA-ACL dentro da mesma plataforma de agentes.
- *Biblioteca de protocolos FIPA:* Para interação entre agentes JADE dispõe uma biblioteca de protocolos prontos para ser usados.
- *Automação de registros:* Registro e cancelamento automático de agentes com o AMS fazendo com que o desenvolvedor se abstraia disso.
- *Serviços de nomes (Naming Service) em conformidade aos padrões FIPA:* na inicialização dos agentes, estes obtêm seus GUID (Globally Unique Identifier) da plataforma que são identificadores únicos em todo o ambiente.
- *Integração:* Mecanismo que permite aplicações externas carregarem agentes autônomos JADE.

Além das características acima citadas que por si só já facilitam muito o desenvolvimento de sistemas multiagentes, Jade possui também algumas ferramentas muito úteis que simplificam a administração da plataforma de agentes e o desenvolvimento de aplicações.

8.2 Regras de comunicação FIPA no JADE

O modelo de plataforma padrão especificado pela FIPA, que pode ser visto na figura ilustrada.

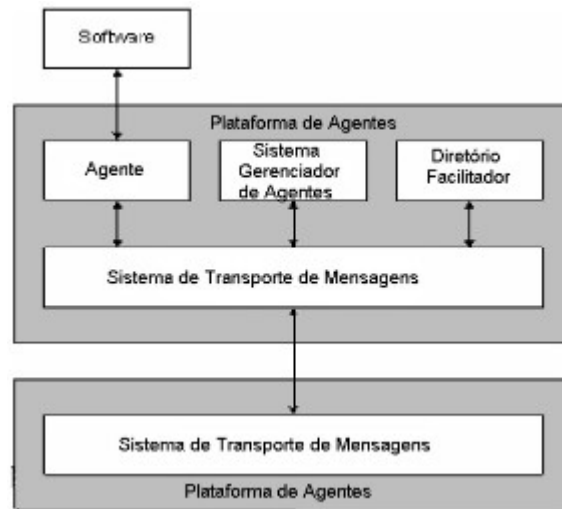


Figura 8.1: Modelo da plataforma de agentes definido pela FIPA

Este modelo pode ser composto por algumas estruturas definidas abaixo:

O *agente*, parte superior esquerda da Figura 9.1, é o agente propriamente dito cujas tarefas serão definidas de acordo com o objetivo da aplicação. Encontra-se dentro da plataforma de agentes (Agent Platform) e realiza toda sua comunicação com agentes através de troca de mensagens e relacionando-se com aplicação externa (software).

O *sistema gerenciador de agentes ou Agent Management System (AMS)*, parte superior central da Figura 9.1, é o agente que supervisiona o acesso e o uso da plataforma de agentes. Apenas um AMS irá existir em uma plataforma. Ele provê guia de endereços (whitepages) e controle de ciclo-de-vida, mantendo um diretório de identificadores de agentes (Agent Identifier - AID³) e estados de agentes. Ele é o responsável pela autenticação de agentes e pelo controle de registro. Cada agente tem que se registrar no AMS para obter um AID válido.

O *diretório facilitador (Directory Facilitator - DF)*, localizado na parte superior direita da Figura 9.1, é o agente que provê o serviço de páginas amarelas (yellow-pages) dentro da plataforma.

Na parte inferior da Figura 9.1 temos o sistema de transporte de mensagens ou *Message Transport System*, também conhecido como canal de comunicação dos agentes (Agent Communication Channel - ACC). Ele é o agente responsável por prover toda a comunicação entre agentes dentro e fora da plataforma. Todos os agentes, inclusive o AMS e o DF, utilizam esse canal para a comunicação.

³AID ou Agent Identifier é uma classe de JADE que atribui identificadores únicos aos agentes.

JADE cumpre totalmente com essa arquitetura especificada. Sendo que, no carregamento da plataforma JADE, o AMS e o DF são criados e o ACC é configurado para permitir a comunicação através de mensagens.

8.3 Agentes em JADE

Para o Jade, um agente é autônomo e independente processo que tem uma identidade e requer comunicação com outras agentes, seja ela por colaboração ou por competição, para executar totalmente seus objetivos (Telecom-Itália (2010)).

Em outras palavras, pode-se concluir que Jade é absolutamente neutro no que diz respeito à definição de um agente, ou seja, ele não limita ou especifica que tipo de agente pode ser construído pela plataforma.

Em termos mais técnicos um agente em Jade é funciona como uma "thread" que emprega múltiplas tarefas ou comportamentos e conversações simultâneas. Esse *agente* Jade é implementado como uma classe Java chamada Agent.

Essa classe Agent atua como uma super classe para a criação de agentes de software definidos por usuários. Ela provê métodos para executar tarefas básicas de agentes, tais como:

Passagens de mensagens usando objetos ACLMessage (seja direta ou multicast); Suporte completo ao ciclo de vida dos agentes, incluindo iniciar ou carregar, suspender e "matar" (killing) um agente; Escalonamento e execução de múltiplas atividades concorrentes; Interação simplificada com sistemas de agentes FIPA para a automação de tarefas comuns de agentes (registro no DF, etc).

O JADE também provê suporte ao desenvolvimento de agentes móveis. Devido a sua própria arquitetura, os agentes podem migrar e clonar-se entre os containeres. Além disso, disponibiliza uma biblioteca (jade.domain.mobility) que contém a definição de ontologias para a mobilidade em JADE, vocabulário com uma lista de símbolos usados e todas as classes Java que implementam essas ontologias.

Do ponto de vista do programador, um agente JADE é simplesmente uma instância da classe Agent, no qual os programadores ou desenvolvedores deverão escrever seus próprios agentes como subclasses de Agent, adicionando comportamentos específicos de acordo com a necessidade e objetivo da aplicação, através de um conjunto básico de métodos, e utilizando as capacidades herdadas que a classe Agent dispõe tais como mecanismos básicos de interação com a plataforma de agentes (registro, configuração, gerenciamento remoto, etc).

Na Figura 9.2 temos a descrição de uma arquitetura interna de um agente em JADE.

Na parte superior temos os comportamentos ativos do agente que representariam as ações/intenções que cada agente tem. O modelo computacional de um agente em JADE é multitarefa, onde tarefas (ou comportamentos) são executadas concorrentemente. Cada

funcionalidade ou serviço provido por um agente deve ser implementado como um ou mais comportamentos. Note que esses comportamentos podem ser diversos uma vez JADE permite uma variedade de comportamentos em um mesmo agente.

Na parte inferior esquerda da Figura 9.2, temos uma fila de mensagens ACL. Todo agente JADE tem essa fila onde decide quando ler as mensagens recebidas e quais mensagens ler.

Ao centro da figura, temos o escalonador de comportamentos e o gerenciador do ciclo de vida. O primeiro é responsável por escalonar a ordem de execução dos comportamentos. O gerenciador de ciclo de vida é o controlador do estado atual do agente.

Como uma característica importante do modelo de agentes JADE é autonomia, cada agente possui uma autonomia que tem a possibilidade de controlar completamente sua thread de execução. O gerenciador de ciclo de vida é meio que os agentes utilizam para determinar seu estado atual (ativo, suspenso, etc). No lado direito da figura, temos os recursos de agentes dependentes da aplicação, e é nesse local serão armazenadas as crenças e capacidades que o agente adquiriu na execução da aplicação.

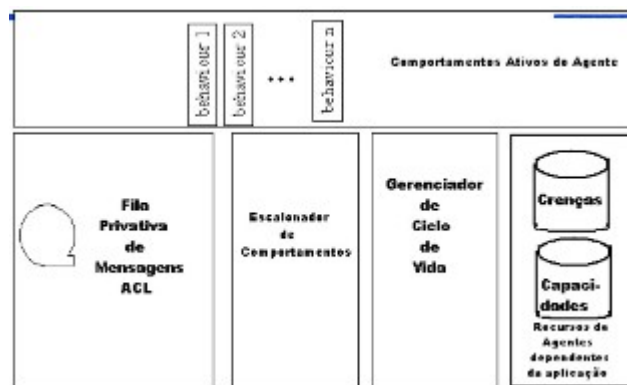


Figura 8.2: Arquitetura interna de uma agente genérico em JADE

8.3.1 Ciclo de vida do Agente

Um agente JADE pode está em um dos vários estados de acordo com ciclo de vida (Agent Platform Life Cycle) das especificações FIPA. Na Figura 9.3, temos a representação do ciclo de vida de um agente definido pela FIPA e seus estados possíveis (FIPA (2010)).

Vale ressaltar que é permitido ao agente executar seus comportamentos (behaviours) apenas quando estiver no estado ativo. Outro fato que também merece ser ressaltado é que se em qualquer dos comportamentos de um agente for chamado o método `Em_espera()`, o agente como um todo e todas suas atividades serão bloqueadas, não apenas o comportamento que chamou o método. Também é importante referencia que o JADE possui métodos que permitem somente a suspensão de um comportamento, evitando o bloqueio total.

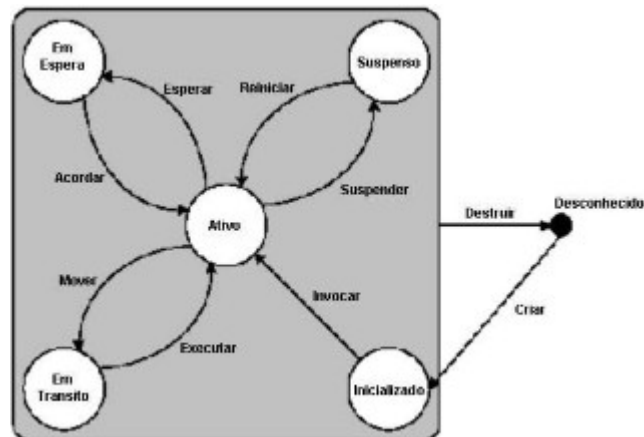


Figura 8.3: Ciclo de Vida de um agente definido pela FIPA

8.4 Troca de Mensagens

Toda a troca de mensagens realizada no JADE é feita através de métodos próprios e com o uso de instâncias da classe `ACLMessage`. Esta classe possui um conjunto de atributos que estão em conformidade com as especificações da FIPA, implementando a linguagem FIPA-ACL. Assim, um agente que pretenda enviar uma mensagem deve instanciar um objeto da classe `ACLMessage`, preenchê-los com as informações necessárias e chamar o método `send()` da classe `Agent`. Caso for receber mensagens, o método `receive()` ou `blockingReceive()` também da classe `Agent` deve ser chamado. Outra meio de enviar ou receber mensagens no JADE é através do uso das classes de comportamentos `SenderBehaviour()` e `ReceiveBehaviour()`. Fato que torna possível que as trocas de mensagens possam ser escalonadas como atividades independentes de um agente.

8.5 Interoperabilidade

Conforme já foi mencionado antes, a arquitetura de JADE é baseada em Java Virtual Machine em diferentes hosts. Para manter as comunicações entre diferentes ambientes JADE utiliza-se do protocolo IIOP⁴ (Internet Inter-ORB Protocol). Para comunicação de agentes na mesma plataforma JADE já utiliza outros meios, como RMI ou via eventos, ou seja, podemos perceber que JADE possui um comportamento variado e distribuído em relação a forma com que realiza sua comunicação, aonde o mecanismo é selecionado de acordo com a situação do ambiente. Isso ocorre com um objetivo único de atingir o menor custo possível de passagens de mensagens.

A seguir é mostrada uma figura que ilustra a interoperabilidade entre os agentes, em que o meio para se estabelecer a comunicação mais eficiente é escolhido de acordo com a localização

⁴Introduzido na segunda especificação de CORBA (Common Object Request Broker Architecture), o IIOP permitiu que CORBA se tornasse uma solução definitiva para interoperabilidade entre objetos que não estão presos a uma plataforma ou padrão específico.

do agente receptor da mensagem conforme foi explicado anteriormente. Na parte inferior da Figura 9.4 existem as formas possíveis de comunicação. O cache local, localizado no ACC (parte central da Figura), armazena as referências dos objetos dos outros containeres. Essas referências são adicionadas ao cache sempre que uma mensagem é enviada.



Figura 8.4: Interoperabilidade entre os agentes

Agora que já foram adquiridos os conhecimentos sobre ontologia, agentes inteligentes/sistemas multiagentes, FIPA e JADE, já podemos partir para a implementação da ferramenta proposta neste trabalho que se baseia em ontologias focadas em teorias sociais e agentes inteligentes.

Capítulo 9

Implementação

Conforme citada no capítulo anterior, uma vez adquiridos os conhecimentos necessários sobre os recursos a serem utilizados, já é possível detalhar a implementação da ferramenta proposta.

Esta ferramenta tem por objetivo criar ontologias de aluno e grupo baseadas em teorias sociais. Essas ontologias são representadas por agentes, que detêm de uma certa inteligência (no caso, artificial) provendo a interação entre aluno/aluno e aluno/grupo. E para alcançarmos este objetivo, a ferramenta será implementada com um framework poderoso na implementação de agentes, o Jade, cujas funcionalidades e características foram apresentadas no capítulo anterior.

O Jade tem todo o suporte para a criação de agentes baseados em ontologias e também possui os protocolos de comunicação do FIPA que permitem a interoperabilidade entre os agentes contidos. Sendo assim, as classes *Aluno* e *Grupo* são os agentes criados com suas respectivas ontologias. Depois serão criadas as classes que irão estabelecer os comportamentos entre os agentes, a comunicação e a interoperabilidade entre os agentes.

Nas próximas seções, serão detalhadas todas as funcionalidades de cada classe que compõem o projeto em si e os algoritmos de cada uma delas poderão ser vistos em "Anexos", bem como o diagrama UML das ontologias de Aluno e Grupo.

e no próximo capítulo serão apresentados os resultados obtidos desta implementação.

Capítulo 10

Conclusões

O presente trabalho teve por objetivo modelar uma ferramenta para EaD baseada em teorias sociais e agentes inteligentes, criando-se todos tipos de agentes que estabelecem uma comunicação entre eles.

O framework Jade que comporta toda a infra-estrutura que promove o desenvolvimento e implementação de ambientes ou sistemas multiagentes voltados para a Educação a Distância. Os sistemas multiagentes não estão restritos somente ao uso em ambientes computacionais de ensino. Eles podem ser utilizados nas mais diversas áreas, tais como: controle de fluxo de rede, compartilhamento de arquivos em rede, simulações de ambientes sociais e etc.

Um aspecto muito importante na construção de sistemas multiagentes é o estabelecimento de mecanismos de comunicação e interoperabilidade entre os agentes, que podem ser obtidos pelos protocolos de comunicação da FIPA.

A organização FIPA surgiu como um esforço na tentativa de padronizar as tecnologias de sistemas multiagentes, que até então os mecanismos de comunicação eram completamente heterogêneos. Ela define todas as características que um sistema multiagente deve implementar e possuir para ser considerado interoperável, ou seja, ter a capacidade de se comunicar os demais agentes presentes na plataforma.

A comunicação dos agentes no JADE é baseada nas regras estabelecidas pela FIPA, o que facilita a comunicação entre os agentes.

O objetivo de se criar agentes *Aluno* e *Grupo* neste trabalho, vem do facto de se propôr que os cursistas sejam mais autônomos no ambiente do seu curso. Que eles possam interagir, troca impressões, trocar material de estudo e sanar as suas dúvidas.

Com a utilização de ontologias baseadas nos perfis sociais dos alunos, é possível identificar os alunos que possuem maiores dificuldades dentro do grupo e encontrar novas metodologias de ensino que possam atender o grau de dificuldade dos alunos.

É possível perceber que a criação desta ferramenta pode diminuir um pouco mais o trabalho do tutor dentro do curso, pois ele não precisará de estar permanentemente levantando questões

e tirando dúvidas aos alunos, pois estes têm a possibilidade de aprenderem um com os outros e o tutor irá intervir quando ele achar necessário.

Espera-se que este trabalho possa servir de grande utilidade para cursos que são ministrados à distância, permitindo a consolidação da EaD.

Apêndice

Referências Bibliográficas

Allahverdi (1999).

Baker, A. (1997). *A java-based agent framework for multiagent systems*. PhD thesis, University of Cincinnati, Department of Electrical & Computer Engineering and Computer Science.

Bassani, P. B. S.; Flores, M. B. e Ritzel, M. (2007). Modelando acessibilidade na web: uma proposta para o desenvolvimento de. *Novas Tecnologias na Educação*, 5(1).

Bradshaw, J. M. (1997). An introduction to software agents. In *Software Agents*, Massachusetts - USA. MIT Press.

BRENNER, W.; RÜDIGER, Z. e WITTIG, H. (1998). Intelligent software. *Springer-Verlag*.

Daniels, H. (2003). *Vygotsky e a Pedagogia*. Loyola Edições, São Paulo, SP, Brasil, 1 edição.

Fensel, D. (2001). Silver bullet for knowledge. *Springer Verlag - International Publisher Science*.

Finin, T.; Labrou, Y. e Mayfeld, J. (1997). Kqml as an agent communication language. *BRADSHAW*, pp. 291–316.

FIPA (2010).

Gómez-Peréz, A. (1999). Evaluation of taxonomic knowledge in ontologies and knowledge bases 12^a. *Workshop ON Knowledge Acquisition, Modeling and Management*.

Lopes, J. G. R. C. (2005). Matching semântico de recursos de computacionais em ambientes de grades com múltiplas ontologias. Master thesis, Instituto de Ciências Exatas, Brasília, DF, Brasil.

Maedche, A. D. (2002). *Ontology Learning for the Semantic Web*. Editora Kluwer Academic.

Moreno, A. O. e Hernández, C. P. (2000). Reusing the mikrokosmos ontology for concept-based multilingual terminology data bases. In *Mikrokosmos*, Athens, Greece. Proceedings of 2^a Conference on Language Resources and Evaluation (LREC 2000).

Puc-Rio (2003). Ontologia. *Puc-Rio, Teses abertas*.

Revista Escola - Durkheim (2008). Émile durkheim - o criador da sociologia da educação. <http://revistaescola.abril.com.br/historia/pratica-pedagogica/criador-sociologia-educacao-423124.shtml>.

Revista Escola - Vygotsky (2008). Lev vygotsky, o teórico do ensino como processo social. <http://revistaescola.abril.com.br/historia/pratica-pedagogica/lev-vygotsky-teorico-423354.shtml>.

RUSSEL, S. e NORVIG, P. (1995). *Artificial Intelligence a Modern*. Prentice-Hall.

Severo, C. E. P.; Passerino, L. M.; Koch, S. H. S.; Maciel, M. e C., G. J. (2009). Uma ontologia para categorias de mediação segundo uma abordagem epistemológica baseada na interação social. *Novas Tecnologias na Educação*, 7(3).

SHOHAM, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, 60(1).

Telecom-Itália (2010).

Vygotsky, L. S. (2001). *A Formação Social da Mente*. Martins Editora, São Paulo, SP, Brasil.