

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE
ÁREA DE CIÊNCIA DA COMPUTAÇÃO

PEDRO BRANDÃO NETO

**MECANISMO DE AUXÍLIO AO PROFESSOR PARA O AMBIENTE
NETCLASS DE ENSINO-APRENDIZAGEM**

São Luís
2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

PEDRO BRANDÃO NETO

**MECANISMO DE AUXÍLIO AO PROFESSOR PARA O AMBIENTE
NETCLASS DE ENSINO-APRENDIZAGEM**

Dissertação de Mestrado apresentada à coordenação do Programa de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão como requisito parcial para a obtenção do título de Mestre em Engenharia de Eletricidade, na área de Ciência da Computação.

Orientador: Prof. Dr. Sofiane Labidi
Co-orientador: Prof. Dr. Ademir Martins

São Luís
2007

Brandão Neto, Pedro.

Mecanismo de auxílio ao professor para o ambiente NetClass de ensino-aprendizagem / Pedro Brandão Neto. – 2007.

121 f.

Impresso por computador (fotocópia)

Orientador : Sofiani Labidi

Co-orientador : Ademir Martins

Dissertação (Mestrado) – Programa de Pós-Graduação em Engenharia de Eletricidade, Universidade Federal do Maranhão.

1. Educação a distância. 2. NetClass. 3. Agentes inteligentes. I. Labidi, Sofiani, orient. II. Martins, Ademir, co-orient. III. Título.

CDU 004.73:37.018.43

A Deus pela minha vida.

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus por tantas bênçãos concedidas em minha vida.

Aos meus familiares em especialmente aos meus pais, incluindo os meus avós, mãezinha e paizinho, pela minha educação, formação para a vida e apoio, incentivo e motivação constantes em todo processo deste trabalho.

Ao meu orientador prof. Sofiane Labidi e co-orientador prof. Ademir Martins pela orientação, confiança, incentivo, dedicação e estímulo para a realização deste trabalho.

À minha querida noiva Priscila Ramos, pelo amor, apoio incondicional, pelas palavras de incentivo nos momentos de angústia e principalmente por tudo que temos compartilhado juntos.

Aos professores, funcionários e alunos do CPGEE.

Aos meus amigos do laboratório de Sistemas Inteligentes.

A Fundação de Amparo à Pesquisa e ao Desenvolvimento Científico e Tecnológico do Maranhão - FAPEMA pelo auxílio financeiro para realização deste trabalho.

De um modo geral, a todos aqueles que de alguma forma contribuíram para a elaboração deste trabalho.

“Como saber se se chegou, se não se sabe onde se quer chegar.”

(Gauthier).

SUMÁRIO

LISTA DE FIGURAS	19
LISTA DE TABELAS	21
LISTA DE ABREVIATURAS E SIGLAS	22
RESUMO	24
ABSTRACT	25
1 INTRODUÇÃO	15
1.1 Organização do trabalho.....	17
2. OBJETIVOS	18
2.1 Objetivo geral.....	18
2.2 Objetivos específicos.....	18
3. JUSTIFICATIVA E RELEVÂNCIA	19
4. EDUCAÇÃO A DISTÂNCIA	24
4.1 Definição da EAD.....	24
4.2 Ambientes de Ensino-Aprendizagem	26
4.2.1 Ferramentas de Comunicação.....	31
4.3 Avaliação	33
4.3.1 Acompanhamento do Aprendiz.....	35
5 INTELIGÊNCIA ARTIFICIAL	38
5.1 Agentes	40
5.2 Sistema MultiAgente	44
5.3 Comunicação entre Agentes	46
5.3.1 Comunicação Indireta.....	47
5.3.2 Comunicação Direta	48
5.4 Plataforma de Agentes	54
5.4.1 JADE (Java Agent DEvelopment Framework)	55
5.4.2 Modelo Padrão da Plataforma JADE	57
5.4.3 Arquitetura Distribuída	58
5.4.4 Agentes na Plataforma JADE	60
5.4.5 Comportamentos	65
5.4.6 Comunicação entre Agentes na Plataforma JADE	68
5.4.7 Ferramentas de Gerenciamento	71

5.5 Sistema Baseado em Conhecimento	72
5.5.1 Formalismo de Representação do Conhecimento	73
5.5.1.1 Sistemas de Produção	74
5.6 JESS (Java Expert System Shell)	76
6 PROPOSTA DE TRABALHO	79
6.1 Mecanismo de Apoio no Acompanhamento do Aprendiz (MAAA)	79
6.1.1 Acompanhamento do Aprendiz no Ambiente NetClass	83
6.1.2 Os Objetivos do MAAA	88
6.1.3 Os Critérios de Avaliação	89
6.1.4 Arquitetura do MAAA	92
6.1.5 A sociedade de Agentes do MAAA	93
6.1.5.1 O Agente Monitor	96
6.1.5.2 O Agente Conversor	97
6.1.5.3 O Agente de Aquisição de Conhecimento	98
6.1.5.4 O Agente de Suporte ao Professor	99
6.1.6 O processo de tomada de decisão do MAAA	101
7. CONSIDERAÇÕES FINAIS	104
REFERÊNCIAS	105
APENDICE A – DEFINIÇÃO DOS COMPORTAMENTOS DO MAAA	111
APENDICE B – CÓDIGO FONTE DOS AGENTES	112
APENDICE C – CÓDIGO FONTE JESS	120

LISTA DE FIGURAS

Figura 1: Elementos de um ambiente virtual.....	29
Figura 2: As interações de um agente genérico com seu ambiente.....	41
Figura 3: Modelo de agente cognitivo.....	43
Figura 4: Fluxo do projeto de um SMA.....	45
Figura 5: Modelo de interação indireta.....	46
Figura 6: Interação entre agentes de forma direta.....	47
Figura 7: Estrutura de Quadro-negro.....	48
Figura 8: Formato de uma mensagem FIPA-ACL.....	54
Figura 9: Modelo padrão de plataforma de agentes definido pela FIPA.....	57
Figura 10: Container e Plataforma.....	59
Figura 11: Plataforma de Agente JADE distribuídas em vários containers.....	60
Figura 12: Arquitetura Interna de um Agente JADE.....	62
Figura 13: Ciclo de vida de um Agente definido pela FIPA (FIPA 2003).....	63
Figura 14: Formas de Comunicações entre Agentes.....	70
Figura 15: Interface Visual do RMA.....	71
Figura 16: Ambiente NetClass.....	81
Figura 17: Estrutura do NetClass.....	82
Figura 18: Módulos de um Curso no ambiente NetClass.....	83
Figura 19: Esquema dos Mecanismos de Avaliação Informal existentes.....	84
Figura 20: Acompanhamento do Aprendiz no NetClass.....	84
Figura 21: Novo Esquema proposto de Avaliação Informal do MAAA.....	86
Figura 22: Esquema do MAAA.....	87
Figura 23: Arquitetura do Modelo do MAAA.....	92
Figura 24: Alimentação da BC.....	93
Figura 25: Diagrama de Classe dos Agentes Inteligentes do MAAA.....	95
Figura 26: Caso de uso do Agente Monitor.....	96
Figura 27: Caso de uso do Agente Conversor.....	97
Figura 28: Diagrama de Seqüência.....	97
Figura 29: Caso de uso do Agente Aquisição de Conhecimento.....	98
Figura 30: Caso de uso do Agente de Suporte do Professor.....	99

Figura 31: Fatos JESS.....	100
Figura 32: Regras JESS.....	101
Figura 33: Organização da BC.....	101
Figura 34: Esboço do Resultado com MAAA.....	102
Figura 35: Mensagem contendo o problema, as possíveis causas e orientação.....	103

LISTA DE TABELAS

Tabela 1: Quadro de representação das Performativas KQML.....	50
Tabela 2: Quadro dos parâmetros reservados de KQML.....	51
Tabela 3: Parâmetros das mensagens FIPA-ACL.....	52
Tabela 4: Atos Comunicativos de FIPA-ACL.....	53
Tabela 5: Métricas da Prova.....	90
Tabela 6: Métricas de contribuição em Fóruns e Chats.....	90
Tabela 7: Métricas das interações do aprendiz com o módulo.....	91
Tabela 8: Métricas de download das lições off-line.....	91
Tabela 9: Métricas de acesso às lições on-line.....	91

LISTA DE ABREVIATURAS E SIGLAS

ACC	Agent Communication Channel
ACL	Agent Communication Language
AID	Agent Identifier
AMS	Agent Management System
IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
BC	Base de Conhecimento
BD	Base de Dados
DF	Directory Facilitator
CSELT	Centro de Estudo e Laboratório de Telecomunicação
CSCCL	Computer Supported Collaborative Learning
DARPA	Defense advanced Research Projects Agency
EAD	Educação a Distância
EDMC	Educação a Distância Mediada por Computador
FIPA	Foundation for Intelligent Physical Agents
GUID	Globally Unique Identifier
IIOF	Internet Inter – ORB Protocol
JADE	Java Agent DEvelopment Framework
JESS	Java Expert System Shell
LGPL	Lesser General Public License
JVM	Java Virtual Machine
LHS	Left-Hand Side
LISP	List Processing
MTP	Message Transport Protocol
MAAA	Mecanismo de Apoio no Acompanhamento do Aprendiz
RHS	Right-Hand Side
RMA	Remote Management Agent
RMI	Remote Method Invocation
SBC	Sistema Baseado em Conhecimento
SMA	Sistemas Multi-Agente

STI	Sistema Tutores Inteligentes
TIC	Tecnologia de Informação e Comunicação
Tilab	Telecomm Italia Lab
WebCT	Web Course Tools
YP	Yellow Pages

RESUMO

Os ambientes de Educação a Distância, atualmente, proporcionam meios de avaliação de o aprendiz. Essa tarefa, de fato, ainda, exige muito esforço e empenho da parte do professor. Portanto, o acompanhamento do aprendiz em ambientes de ensino-aprendizagem a distância, obtida através da observação das atividades dos aprendizes, apresenta problemas referentes à carência de mecanismos que ofereçam um suporte eficaz ao professor na orientação para o aprendiz. Esses mecanismos existentes somente apresentam resultados ao professor sem fazer qualquer inferência, desta maneira, o próprio professor precisa analisar manualmente esses resultados em busca de subsídios para fazer uma orientação precisa, mostrando as possíveis causas e direcionando o aprendiz para o caminho correto. Entretanto, para dotar um mecanismo com capacidade de inferência faz-se necessário capturar o conhecimento de como o professor realiza o acompanhamento do aprendiz para, posteriormente, transferir este conhecimento para uma base de conhecimento com capacidade de inferência. O conhecimento necessário para o professor realizar o acompanhamento do aprendiz em um ambiente de ensino-aprendizagem são os critérios de avaliação que são definidos pela coordenação pedagógica ou professor do curso da instituição de ensino. No entanto, esses critérios de avaliação são genéricos e não padronizadas dificultando, portanto, o consenso do mesmo; foi realizado um levantamento teórico para a definição dos critérios de avaliação no ambiente NetClass. Esta pesquisa gerou o conhecimento sobre o desenvolvimento de um modelo de um mecanismo denominado de Mecanismo de Apoio no Acompanhamento do Aprendiz (MAAA) o qual é aplicado ao ambiente NetClass de ensino-aprendizagem.

Palavras-chave: Inteligência Artificial, Agentes Inteligentes, Educação a Distância, Ambiente Colaborativo e Acompanhamento do Aprendiz.

ABSTRACT

Distance Education environments, nowadays, provide means for learner evaluation. This task, still, requires effort of the teacher. So, the learner accompaniment in distance teaching-learning environments, obtained through the observation of the learner activities, it presents referring problems to the lack of mechanisms that offer an effective support to the teacher in the orientation for the learner. These existent mechanisms only present results to the teacher without to do any inference, like this, self teacher needs to analyze those results manually in search of subsidies to do a precise orientation, showing the possible causes and addressing the learner for the correct road. However, to dower a mechanism with inference capacity is necessary to capture the knowledge of as the teacher fulfills the leaner accompaniment for, later, to transfer this knowledge for a knowledge base with inference capacity. The necessary knowledge for the teacher to realize the learner accompaniment in a teaching-learning environment is the evaluation criterions that are defined by course pedagogic coordination or teacher of the teaching institution. However, those evaluation criterions are generic and not standardized hindering, therefore, the consent of the same; was realized a theoretical rising for the definition of the evaluation criterions in the NetClass environment. This research generated the knowledge on the development of a model denominated Learner Accompaniment Support Mechanism based on techniques of Artificial Intelligence, which is applied to the NetClass environment of teaching-learning.

Keywords: Artificial Intelligence, Intelligent Agent, Distance Education, Collaborative Environment and Learner Accompaniment.

1 INTRODUÇÃO

Este trabalho apresenta um mecanismo de auxílio ao professor no acompanhamento do aprendiz no ambiente NetClass de ensino-aprendizagem, no qual é baseado em técnicas da Inteligência Artificial. O mecanismo desenvolvido é uma proposta inserida no contexto do Projeto NetClass que aborda meios para melhorar a Educação a Distância (EAD).

A Inteligência Artificial provê técnicas para o desenvolvimento de programas de computador que realizam várias tarefas de domínios diferentes objetivando, portanto, simular um comportamento mais inteligível na resolução de problemas (NILSSON, 1986, pág. 3; POPOVIC et al., 1994, pág. 40). Portanto, para um programa de computador ter a capacidade de resolver um problema de um determinado domínio a Inteligência Artificial fornece um conjunto de formalismo para modelar problemas e também técnicas para resolvê-lo (LUGER, 2002, pág. 40).

Atualmente, técnicas de Inteligência Artificial têm sido amplamente aplicadas em modelos instrucionais para oferecer modelos interativos e dinâmicos de educação com intuito de facilitar ao máximo todo o processo de ensino-aprendizagem (HARRER et al., 2006, pág. 1).

No final do século passado, a difusão e eclosão da internet possibilitaram o surgimento de ambientes virtuais que simulam e promovem a EAD (MEHLECKE et al., 2003, pág. 8; ALMEIDA, 2003, pág. 3). Iniciou-se, logo em seguida, o aperfeiçoamento de técnicas e de ferramentas associadas a esses ambientes de ensino-aprendizagem advindas de esforços de pesquisadores tanto na área de Educação a Distância como na área de Tecnologia de Informação e Comunicação (TIC), trazendo, portanto, melhorias relevantes no processo de educação neste cenário (DIMITRACOPOULOU et al., 2003, pág. 3; COLLIS, 1999; pág. 85).

A finalidade desta dissertação é fornecer um modelo de um mecanismo, baseado em arquitetura multiagentes, para dar suporte ao professor no acompanhamento do aprendiz no ambiente NetClass de ensino-aprendizagem colaborativa.

A Educação a Distância (EAD) é definida como uma forma de ensino que possibilita a auto-aprendizagem com a mediação de recursos didáticos sistematicamente organizados, apresentados em diferentes suportes de informação, utilizados isoladamente ou combinados e veiculados pelos meios de comunicação, dando destaque a elementos de abertura à democratização do ensino e autonomia do indivíduo (KRAMER, 1990).

Os ambientes de ensino à distância utilizam, ultimamente, o paradigma de colaboração com intuito de incentivar os aprendizes a interagirem entre si, melhorando, assim, a comunicação entre o aprendiz e o professor (JERMANN et al., 2001, pág. 3). Segundo Jaques (2000), quando os alunos interagem, eles se sentem mais motivados e engajados nas atividades do curso e, por isso, obtêm melhores resultados nos estudos.

Para garantir a eficiência da educação a distância, como evidencia a pesquisadora Polak (2002, pág. 66), é necessário assegurar a existência de mecanismos de monitoramento e avaliação constantes do sistema educacional e dos recursos utilizados. Os ambientes que promovem a aprendizagem colaborativa mediada por computador disponibilizam, atualmente, vários meios de interação para que os aprendizes possam compartilhar idéias, intercambiar informações em prol da aprendizagem. Como consequência do registro dessas interações, há um número considerável de informações geradas no ambiente do professor. A observação dessas informações caracteriza o processo de acompanhamento do aprendiz denominado avaliação informal. Essa avaliação é dificultada pela perda do contato face a face entre o professor e o aprendiz.

Muitos ambientes que dão suporte a aprendizagem colaborativa apresentam deficiências no processo de acompanhamento do aprendiz pela falta de mecanismos adequados para o auxílio ao professor no apoio à tomada de decisão – pois esses somente auxiliam o professor na análise dos dados quantitativos – e pela natureza intrínseca dos critérios de avaliação. O presente trabalho descreve uma nova abordagem no processo de acompanhamento do aprendiz objetivando proporcionar um suporte flexível na orientação e/ou direcionamento do aprendiz, ao mesmo tempo em que promove a redução da sobrecarga de informações no ambiente do professor. Este processo será realizado por um mecanismo que irá aproveitar de uma forma mais eficiente o crescente volume de dados no ambiente do professor, realizando, com a ajuda de agentes artificiais (reativos e cognitivos), o processo de acompanhamento do aprendiz e a posterior orientação através de inferências computacionais fornecendo assim um suporte qualitativo ao invés de apenas apresentar uma análise quantitativa dos dados.

1.1 Organização do trabalho

Esta dissertação está organizada em sete capítulos. O Capítulo 1 é a introdução. No Capítulo 2 é descrito os objetivos geral e específicos. No Capítulo 3 encontra-se o cenário que motivou a realização deste trabalho. O Capítulo 4 apresenta uma visão geral em torno dos conceitos, tecnologias e ferramentas da Educação a Distância. No Capítulo 5 apresenta-se uma visão geral da Inteligência Artificial, Sistemas Multiagentes e a plataforma JADE. O Capítulo 6 é relatado o processo proposto de acompanhamento do aprendiz no ambiente NetClass de ensino-aprendizagem utilizando o Mecanismo de Apoio no Acompanhamento (MAAA) do Aprendiz. O último capítulo apresenta nossas conclusões.

2. OBJETIVOS

2.1 Objetivo geral

O objetivo geral do presente trabalho é desenvolver um modelo de um mecanismo para auxiliar o professor no acompanhamento do aprendiz no ambiente NetClass de ensino-aprendizagem.

2.2 Objetivos específicos

Com a finalidade de alcançar o objetivo pretendido, busca-se-á os seguintes objetivos específicos:

- (a) Reunir informações das ações e atividades do aprendiz e representar essas informações na base de conhecimento de um agente cognitivo através de um formalismo de representação de conhecimento;
- (b) Capturar o conjunto de regras que representa o conhecimento para o professor efetuar o acompanhamento do aprendiz;
- (c) Elaborar a mensagem contendo a orientação ao aprendiz gerada pelo MAAA.

A realização deste trabalho refere-se a descrição de um modelo para dar suporte ao professor no acompanhamento do aprendiz.

3. JUSTIFICATIVA E RELEVÂNCIA

Desde a década de setenta, pela utilização de Sistemas Tutores Inteligentes (STI), o computador vem sendo utilizado no suporte às atividades em educação e treinamento (HARRER et al., 2006, pág. 1). A tecnologia STI é um campo de pesquisa multidisciplinar, que utiliza técnicas de Inteligência Artificial, objetivando a reprodução computacional do comportamento de um tutor humano competente, capaz de adaptar suas estratégias pedagógicas de ensino, de acordo com o perfil do estudante. Esses sistemas, inicialmente, foram baseados no que é nomeado de modelo de ensino um-para-um (MOREIRA, 2004). Esse modelo, no entanto, tem, desde o final dos anos oitenta, sofrido críticas no sentido de que o papel do computador deveria ser o de cooperar com os aprendizes para facilitar o processo de aquisição de conhecimento em vez de, somente, representar o papel de um professor.

Em resposta aos STIs clássicos, criou-se a categoria denominada Ambiente Interativo de Aprendizagem. Nessa categoria, o estilo da interação sofre mudanças para permitir ao aprendiz uma participação mais ativa, pois o controle da interação passa a ser compartilhado entre o sistema e o aprendiz.

O desenvolvimento da Internet e as Tecnologias de Informação viabilizaram a modalidade de aprendizagem colaborativa em sistemas computacionais de ensino-aprendizagem, denominada de aprendizagem colaborativa suportada por computador (CSCL – Computer Supported Collaborative Learning) (JAQUES, 2001). Segundo Labidi et al. (2003), um ambiente de aprendizagem colaborativa suportada por computador (CSCL) traz inúmeras vantagens, tais como: a competitividade sadia entre os aprendizes, compartilhamento de recursos, aumento na produtividade de aprendizagem, desenvolvimento das habilidades sociais e estímulo ao aprendizado.

Nos ambientes de ensino-aprendizagem colaborativa que promovem a Educação a Distância existem formas de realizar o acompanhamento do

aprendiz. As formas principais de avaliação são: diagnóstica, formativa e somativa. No contexto da proposta deste trabalho é usada a avaliação formativa.

Segundo Perrenoud (1999), a avaliação formativa é “toda prática de avaliação contínua que pretenda melhorar as aprendizagens em curso, contribuindo para o acompanhamento e orientação dos alunos durante todo seu processo de formação. É formativa toda a avaliação que ajuda o aluno a aprender e a se desenvolver, que participa da regulação das aprendizagens e do desenvolvimento no sentido de um projeto educativo”.

Nesses ambientes de ensino-aprendizagem colaborativa suportado por computador as informações são geradas rapidamente pelos meios de comunicação existentes como, por exemplo, chats, fóruns de discussões, mural, e-mail dentre outros (LARIOS et al., 2006, pág. 2). Essas ferramentas de comunicação são utilizadas para dar suporte ao aprendiz na compreensão de atividades através de troca de mensagens textual com professor ou outro aprendiz. Toda essa interação é registrada, e isso causa, então, um acúmulo de dados no ambiente à disposição do professor. Por sua vez, o professor tem que empregar várias horas confrontando dados para extrair informações relevantes no seu ambiente levando em consideração um determinado conjunto de informações pré-definido (critérios de avaliação) para efetivação do acompanhamento do aprendiz e a posterior tomada de decisão. Muitas das vezes, o professor não chega, ou melhor, não inferi a devida conclusão. Essa assertiva foi observada e comprovada no ambiente NetClass de ensino-aprendizagem colaborativa pela realização de um estudo de caso. Portanto, torna-se evidente a necessidade do desenvolvimento de um mecanismo capaz de auxiliar o professor na orientação ao aprendiz utilizando o conhecimento do professor. Essa orientação é o apoio à tomada de decisão que é alcançada, de forma espontânea, com a relação das informações do aprendiz que se encontram no ambiente do professor.

Otsuka et al. (2002), explicam que os ambientes de suporte a aprendizagem colaborativa na EAD possuem ferramentas de comunicação que são projetadas para possibilitarem a realização de atividades baseadas na construção colaborativa de conhecimentos e mecanismos de acompanhamento das interações do aprendiz durante o desenvolvimento destas atividades. A observação realizada pelo professor das informações geradas pelas ações do aprendiz no seu ambiente é conhecida, tanto na modalidade de Educação presencial quanto na Educação a Distância, como avaliação informal. Essa avaliação informal possibilita ao professor “perceber” o comportamento e/ou desempenho do aprendiz pela análise dos registros de participações do mesmo, tais como: atividades desenvolvidas, registros de interações e colaboração entre os aprendizes.

Pesquisas como de a Romani (2002), Musa (2001), Shen (2001) demonstram que existe uma variada gama de ambientes que provêem mecanismo(s) para avaliação e/ou acompanhamento do aprendiz, mas, apesar da existência desses mecanismos, ainda não é o suficiente para o acompanhamento efetivo, pois essa tarefa ainda demanda muito tempo e trabalho por parte do professor ao longo do curso. Deste modo, observa-se que esses mecanismos somente auxiliam o professor apresentando os dados colhidos utilizando tabelas ou gráficos. O professor deve, então, analisar esses dados, manualmente, a fim de identificar informações relevantes; por isso, requer esforço e empenho por parte do mesmo. Posteriormente, após a observação realizada das informações do aprendiz, o professor pode elaborar suas conclusões.

Com objetivo de encontrar meios para aprimorar esse processo de acompanhamento do aprendiz, pesquisadores como, por exemplo, Silva et al., (2001), Zaiane et al., (2001), entre outros, vêm explorando novas tecnologias computacionais com a finalidade de fornecer um meio satisfatório que automatiza e diminua essa sobrecarga de informações geradas no ambiente do

professor. De acordo com a literatura pesquisada, podem-se citar as seguintes tecnologias empregadas no processo de acompanhamento do aprendiz:

- A tecnologia de agentes de software que analisam e filtram as participações dos aprendizes nos cursos (JAQUES et al., 2000).
- Uma Arquitetura Multiagente para monitorar os mecanismos de comunicação em um ambiente telemático de ensino (JAQUES et al., 1998).
- Aplicações de técnicas de mineração de dados (classificação, agrupamento e associação) para extração e inferências de padrões (CRUZ, 2003). Essa técnica está sendo aplicada, recentemente, no acompanhamento do aprendizado do aprendiz fornecendo suporte mais elaborado na análise dos dados.

A utilização das técnicas supracitadas, de acordo com a pesquisa realizada por Otsuka (2002), torna-se inadequada, pois é preciso pré-definir um conjunto de informações relevantes e critérios de avaliação no desenvolvimento de um curso. Essa dificuldade encontrada é inerente ao processo de avaliação informal na Educação a Distância, porque cada professor possui uma maneira peculiar de observar e avaliar o aluno. Em vista disso, essas técnicas não atendem perfeitamente a necessidade, embora tenham representado um grande avanço.

Frente a essa problemática, o presente trabalho propõe um modelo, com a utilização de uma sociedade de agentes reativos e cognitivos, para a efetivação do processo de acompanhamento do aprendiz, conforme a descrição a ser realizada no capítulo 6.

Portanto, a contribuição deste trabalho é o desenvolvimento de um modelo para auxiliar o professor no acompanhamento do aprendiz. Este

modelo automatiza a tarefa do próprio professor analisar manualmente os resultados, provenientes dos mecanismos gerados em seu ambiente, a procura de informações do aprendiz para a elaboração de uma orientação coerente. Então, com o desenvolvimento do modelo do MAAA o professor obtém uma orientação do aprendiz através de inferências computacionais descrevendo os possíveis problemas e como solucioná-los.

4. EDUCAÇÃO A DISTÂNCIA

Neste Capítulo será apresentada uma visão geral sobre Educação a Distância (EAD). As características pertinentes dos ambientes de ensino-aprendizagem serão descritas; realizou-se uma revisão da literatura no contexto atual do processo de avaliação para possibilitar o entendimento de como funciona esse processo de acompanhamento do aprendiz em ambientes de ensino-aprendizagem.

4.1 Definição da EAD

A sociedade está em constante renovação, e para que o indivíduo possa acompanhar as mudanças tecnológicas é preciso que este faça da aprendizagem um processo contínuo em busca de mudanças permanentes no seu comportamento. Assim, a tecnologia contribui na aprendizagem, na medida em que possibilita o acesso rápido ao conhecimento, e também permite a composição e a recomposição de dados, informações, argumentos e idéias, situando o professor como mediador entre o conhecimento, a cultura sistematizada, a condição de aprendizagem do aprendiz, beneficiando o entendimento do processo de ensinar. Desta forma, o processo ensino-aprendizagem a distância promove a construção cooperativa do conhecimento, o desenvolvimento da consciência crítica e o favorecimento das soluções criativas para os novos problemas que se impõem (SILVA, 2001).

A Educação a Distância (EAD) é uma modalidade educativa que surge como alternativa para democratizar o saber (INHAN, 2002). Portanto, deve ser compreendida como uma modalidade não tradicional, que dispõem de métodos, técnicas e recursos organizados sistematicamente que contribui para a democratização do saber.

Segundo Moore et al. (1996), a história da EAD é dividida em três gerações. A primeira geração foi baseada em correspondências e estudos

independentes que iniciou no final do século XVIII com o grande desenvolvimento do serviço postal na Europa. A segunda iniciou-se em 1970, com a inclusão da teleconferência, da televisão educativa, das fitas de vídeo e áudio. A terceira geração, marcada pelo uso dos computadores, redes e multimídia digital, tem possibilitado aos pesquisadores a investigação e exploração de meios e métodos objetivando melhorias significativas no ato de aprender. Essa terceira geração, a atual, no ponto de vista da pesquisadora Iahn (2002), trouxe os benefícios da interatividade, semelhante à educação tradicional, e a possibilidade dos aprendizes estudarem a qualquer hora e em qualquer lugar desde que estejam munidos das ferramentas tecnológicas necessárias. Portanto, um fato interessante é que a EAD não apareceu com o desenvolvimento das novas tecnologias, ao contrário de que algumas pessoas acreditam.

Segundo Aretio (1994), a Educação a Distância Mediada por Computador (EDMC) é um sistema tecnológico de comunicação bidirecional, que pode ser massivo e que substitui a interação pessoal, na sala de aula, de professor e aluno, como meio preferencial de ensino, pela ação sistemática e conjunta de diversos recursos didáticos e pelo apoio de uma organização e tutoria que propiciam a aprendizagem independente e flexível dos alunos. Portanto, com a eclosão das tecnologias de comunicação proporcionou a EAD, ao professor e ao aluno, uma comunicação bidirecional, no lugar em que o aprendiz participa mais efetivamente e criticamente do processo de ensino-aprendizagem, surgindo, então, os Ambientes Virtuais de Ensino-Aprendizagem para apoiar cursos à distância na internet.

Esses ambientes de ensino-aprendizagem fornecem meios de centralizar as informações relativas aos cursos e possibilita, também, a geração de dados importantes no acompanhamento do aprendiz. Eles disponibilizam ferramentas necessárias para o professor gerenciar e criar um curso a distância.

4.2 Ambientes de Ensino-Aprendizagem

Os ambientes de aprendizagem virtual são ambientes eletrônicos que fornecem elementos apropriados de aprendizagem para os aprendizes encontrar e trocar seus pensamentos, produtos e idéias independentes de qualquer limitação de espaço e tempo (AKAR et al., 2004).

Segundo Kemczinski (2005), os ambientes virtuais de aprendizagem são sistemas computacionais destinados ao suporte de atividades de ensino-aprendizagem mediados pelas Tecnologias de Informação e Comunicação (TIC). Estes ambientes integram múltiplas mídias, linguagens e recursos; apresentam informações de maneira organizada; desenvolvem-se no tempo, ritmo de aprendizagem e espaço em que cada participante se localiza, de acordo com uma intencionalidade explícita e um planejamento prévio denominado design educacional, o qual constitui a espinha dorsal das atividades a serem realizadas, sendo revisto e re-elaborado continuamente no andamento da atividade.

Existem, nos dias atuais, ambientes de ensino-aprendizagem colaborativa disponíveis para a utilização da sociedade que podem ser gratuitos ou não, a saber:

- WebCT: O WebCT foi desenvolvido e comercializado pelo Departamento de Ciência da Computação da University of British Columbia; dá suporte à criação de cursos sofisticados baseados na Internet para usuários leigos na área de computação. Disponibiliza uma gama variada de ferramentas de comunicação tanto síncronas quanto assíncronas (WEBCT, 2000).
- AulaNet: O AulaNet é um ambiente de aprendizado que promove a aprendizagem cooperativa baseado na Web que foi desenvolvido no Laboratório de Engenharia de Software do Departamento de Informática

da PUC-Rio, cujo projeto iniciou em 1997. Ele dá suporte à criação, manutenção, administração e assistência de cursos através da Internet para pessoas sem conhecimentos técnicos de computação. Seu principal objetivo é permitir a utilização da Internet como um meio viável para a educação, tentando reutilizar material didático já existente e incentivando a criação de comunidades de conhecimento (FUCKS, 2000; AULANET, 2000).

- TelEduc: O TelEduc é um ambiente de suporte ao ensino-aprendizagem à distância, para criação, participação e administração de cursos na web; foi desenvolvido em 1997 pelo Núcleo de Informática Aplicada à Educação (NIED) e que vem sendo continuamente aprimorado juntamente com o Instituto de Computação, ambos da Universidade Estadual de Campinas (UNICAMP). Dentre as características desse ambiente pode-se citar a possibilidade de contextualização das atividades realizadas pelos alunos, disponibilidade de ferramentas de comunicação tanto síncronas quanto assíncronas, além de um conjunto de ferramentas adicionais que permitem ao professor a autoria e o gerenciamento de um curso ministrado por meio do ambiente (TELEDUC, 2000).
- Eureka: é um ambiente de aprendizagem colaborativa à distância por meio da Internet destinado ao estabelecimento de comunidades virtuais de estudo. Ele integra diversas funções em um mesmo ambiente, permitindo a comunicação e o estudo colaborativo.
- HyCLASS: é um sistema colaborativo educacional projetado para permitir que vários estudantes, em localidades distantes, compartilhem um espaço virtual tridimensional. Este ambiente pode ser utilizado para realização de experimentos virtuais e procedimento de tarefas criativas: os estudantes podem criar objetos dinamicamente, modificar e ativar/desativar suas propriedades e comportamentos.

Os ambientes de ensino-aprendizagem colaborativa os quais proporcionam o aprendizado colaborativo têm como objetivo ensinar os alunos a pensar, a raciocinar, a resolver problemas trocando idéias e informações com seus colegas. Desta maneira, os alunos obtêm melhores resultados nos seus estudos, pois ao colaborarem esses trocam conhecimento e, por isso, se sentem mais engajados.

Para tanto, para possibilitar essa interação entre aprendizes os ambientes disponibilizam ferramentas de comunicação para apoiar o processo de comunicação através da rede. Portanto, as ferramentas de comunicação existentes nos ambientes virtuais viabilizam as trocas de mensagens entre os aprendizes, tais como: correio eletrônico, bate-papo, videoconferência, listas de discussão, mural de recados, agendas, etc. Sendo assim, os ambientes supracitados possuem meios de comunicação com a intenção de promover a colaboração. Desta forma, há uma sobrecarga de informação textual das atividades que o professor precisa analisar para acompanhar o andamento do aprendiz durante o transcorrer do curso.

Nos últimos anos, esses ambientes colaborativos se consolidaram devido o crescimento e o desenvolvimento da internet ocasionada pela TIC, e estão, a cada dia, se aperfeiçoando utilizando técnicas apoiadas por Inteligência Artificial tendo em vista melhorias no ato de aprender (ZAIANE et al., 2001).

A figura 1 apresenta alguns elementos que, freqüentemente, compõem essa modalidade de educação e trabalham de forma integrada com o auxílio de tecnologia, que são: tutor, conteúdo pedagógico, suporte administrativo, suporte técnico, sistema de gerenciamento de aprendizagem, professor, monitor, administradores e mídias (KEMCZINSKI, 2005 apud WEBSCHOOL, 2002).

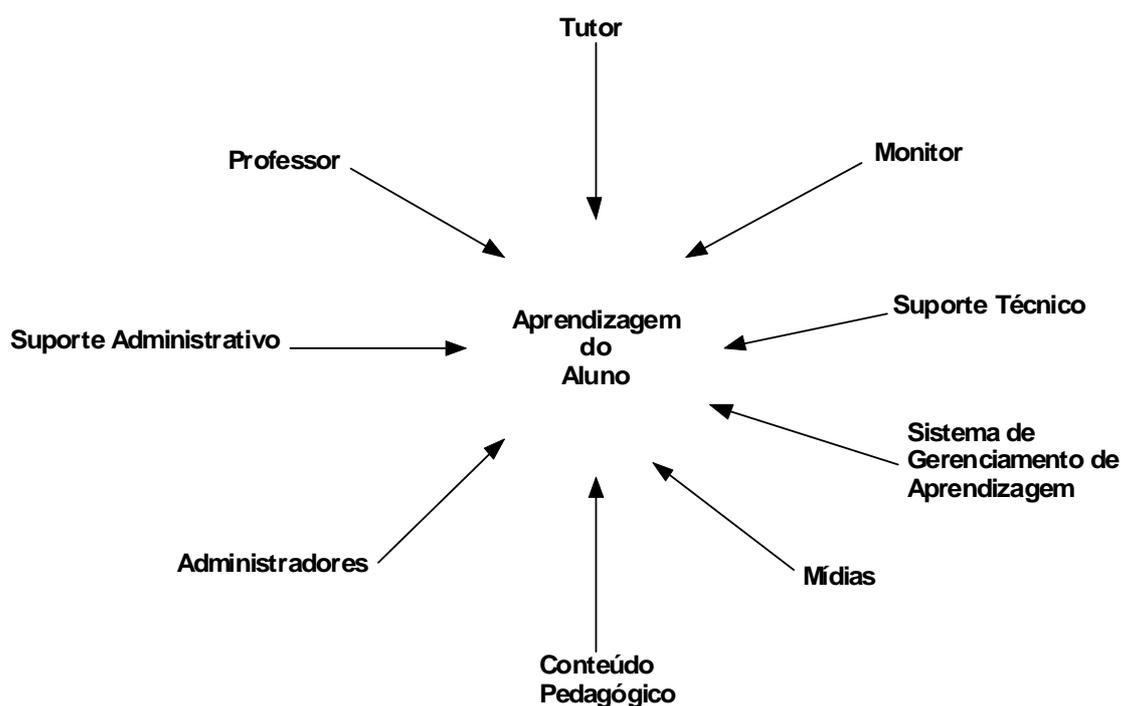


Figura 1: Elementos de um ambiente virtual
Fonte: Webschool (2002)

- aprendizagem do aluno: o aluno é o elemento fundamental no processo de ensino-aprendizagem, desta maneira, esse elemento é o centro de todo processo; para satisfazer suas necessidades deve ser a meta de todo o ambiente;
- professor: o sucesso do ambiente depende fundamentalmente do professor, que tem as mesmas responsabilidades do ensino presencial: selecionar o conteúdo do curso, compreender as necessidades dos estudantes e avaliar a aprendizagem. Demais, deve adaptar o seu estilo de ensino, dominando as mídias adotadas e, por último, deve atuar como facilitador e provedor do conteúdo;
- tutor: tutor é o indivíduo que orienta e reorienta a aprendizagem dos alunos, ajuda no esclarecimento de suas dúvidas identifica dificuldades, sugere novas leituras ou atividades. Portanto, o papel do tutor é assistir ao professor, esclarecendo dúvidas dos alunos, corrigindo exercícios, interagido freqüentemente com ele;

- monitor: nem sempre conhece os conteúdos ensinados. Seu papel equivale aos olhos e ouvidos locais do professor. Opera os equipamentos de uma sala remota, aplica atividades e avaliações e motiva os alunos dando um toque mais pessoal e humano;
- suporte técnico: é responsável por todos os aspectos técnicos relacionados com o sistema, tais como: operação e manutenção de equipamentos, configuração de softwares, incluindo também a criação de material didático, nos aspectos de programação, projeto visual, concepção pedagógica e etc.;
- sistema de gerenciamento de aprendizagem: geralmente é composto por módulos, responsáveis por controlar o acesso ao curso, gerenciar matrículas, dar suporte à comunicação, registrar frequência, mediar a interação entre professor-aluno, aluno-aluno e administração-aluno;
- mídias: são os meios de comunicação através dos quais são trocadas informações entre as partes integrantes do sistema, dentre eles: e-mail, chat e fóruns;
- conteúdo pedagógico: o material didático é o conteúdo elaborado e produzido em diferentes formatos, tais como: páginas web, arquivos de texto, imagens, áudio e vídeo, etc. Um aluno em um ambiente virtual estuda por conta própria, a organização do “espaço” pedagógico muda e as “aulas” passam a ser lições/módulos/capítulos, contidos no material didático. Ainda que elas estejam organizadas em certa ordem, o aluno não está física e temporalmente atado àquela seqüência ou ritmo de estudo, como ele estaria em uma educação presencial, com lições/módulos/capítulos estruturados em aulas. As “aulas” em EAD estão organizadas dentro de um espaço pedagógico chamado material didático (IAHN, pág. 29, 2002).

- administradores: os administradores são os responsáveis pelo gerenciamento de um ambiente virtual; decidem sobre equipamentos, formatos, contratações, prazos, prioridades, etc;
- suporte administrativos: é responsável pelas matrículas, logísticas da distribuição do material, aquisição do material didático e suprimento em geral, controle de direitos autorais, processamento de notas, etc.

Portanto, os ambientes colaborativos atuais de EAD reúnem e fazem disponíveis ferramentas de comunicação para dar suporte ao processo de ensino-aprendizagem, tais como: correio eletrônico, bate-papo, videoconferência, listas de discussão, mural de recados, agendas, entre outros. Essas ferramentas serão abordadas a seguir, na próxima subseção.

4.2.1 Ferramentas de Comunicação

As ferramentas de comunicação disponibilizadas em um ambiente de ensino-aprendizagem colaborativa possibilitam ao aprendiz interagir em seu grupo ou com outros grupos e outras entidades. Portanto, as entidades de um ambiente colaborativo virtual trocam idéias e compartilham conhecimentos em benefício de o ensino via meios de comunicação.

Segundo Zaina (2002) as diversas ferramentas interativas utilizadas pelos aprendizes podem ser divididas nas seguintes categorias: ferramentas síncronas e assíncronas. Nas ferramentas síncronas a comunicação é em tempo real em que as entidades participantes estão presentes em um mesmo espaço de tempo. Por outro lado, nas ferramentas assíncronas a comunicação se diferencia por não ter a necessidade dos participantes envolvidos estarem conectados no mesmo espaço de tempo, ou seja, o emissor e receptor não precisam estar conectados no mesmo instante. Conforme Silva (2002) as principais ferramentas de comunicação de um ambiente, são:

- fórum: o fórum é uma ferramenta assíncrona muito importante porque possibilita o compartilhamento de conhecimento, promove e registra discussões de idéias entre professor e aprendizes de um curso. Tanto o professor quanto o aprendiz podem enviar mensagens, todavia, essas mensagens não são privativas, podendo ser acessadas, vistas e debatidas por todos, mesmo quando endereçadas a um determinado aprendiz.
- bate-papo (chat): escrito, imagem e áudio – é um encontro virtual entre duas ou mais pessoas, de maneira síncrona, que ocorre em um canal onde se pode conversar por escrito ou verbalmente e ainda trocar arquivos. Os ambientes virtuais geralmente desenvolvem sua própria ferramenta de bate-papo.
- correio eletrônico: as mensagens enviadas pelo correio eletrônico são privativas. A importância do serviço de correio eletrônico (e-mail) se dá pelo fato de ele ser restrito somente aos aprendizes de um curso e não utilizar o e-mail verdadeiro do aprendiz, permitindo maior privacidade ao aprendiz que esteja utilizando o curso. Essas mensagens enviadas são armazenadas no servidor de e-mail para posterior acesso do destinatário. Um remetente ao compor uma mensagem pode integrar imagens, sons, vídeos, dentre outros e enviar a um destinatário.
- quadro de comunicações: é uma ferramenta de desenho que permite aos alunos se comunicar com outros usuários logados digitando texto, desenhando objetos, inserindo gráficos e fazendo modificações. Trata-se de uma ferramenta especialmente útil para desenhar e visualizar diagramas em tempo real.
- agenda/cronograma de atividade: como na forma clássica, as agendas e cronogramas dão os indicadores de datas, prazos, resultados e , enfim

de toda a previsão das atividades, com a finalidade de levar a termo o compromisso da formação entre professores/tutores e aprendizes.

A maioria da interação realizada pelo o aprendiz em um ambiente de ensino-aprendizagem é por meio das ferramentas de comunicações. Dessa forma, essas interações acabam gerando uma sobrecarga de informação para o professor tornando, portanto, uma tarefa muito trabalhosa. O professor é quem analisa toda a informação textual gerada pelos aprendizes de um curso a distância com o intuito de avaliar e acompanhar o aprendiz.

Em geral, nota-se que os ambientes virtuais disponíveis fornecem um conjunto de mecanismos para permitir um acompanhamento do aprendiz. Segundo Akar et al. (2004), os mecanismos existentes nos ambientes virtuais, nos últimos anos, vêm adicionando recursos com a finalidade de melhorar a avaliação e acompanhamento do aprendiz porque os professores continuam tendo dificuldades para acompanhar e avaliar os aprendizes de um curso e ter meios de auxiliá-los.

Portanto, pesquisadores da área da Informática na Educação e Computação investigam mecanismos para a realização da avaliação e acompanhamento do aprendiz para suprir, assim, a ausência do contato físico entre o aprendiz e o professor.

4.3 Avaliação

A avaliação é imprescindível no processo de ensino-aprendizagem em ambientes virtuais, pois é através dela que o professor pode acompanhar o andamento do aprendiz e buscar meios de detectar problemas para fazer uma orientação ao aprendiz. Sendo uma das principais etapas, a avaliação não deve ser desvinculada das demais no processo de ensino-aprendizagem. Nota-se que a avaliação não é uma atividade final e sim uma atividade constante

que tem como propósito de determinar se os objetivos de o aprendizado foram alcançados.

Segundo Caleffe (2002), o processo ensino-aprendizagem é uma atividade humana de primeira grandeza com um protagonista principal, o sujeito que aprende, e com outro não menos importante que existe precisamente para facilitar essa aprendizagem, o professor. Nela, obviamente, são igualmente necessárias a medida, a valorização e a tomada de decisão, ou seja, a avaliação.

"Avaliação é a coleta sistemática de evidências por meio das quais determinam-se mudanças que ocorrem nos alunos e como elas ocorreram. Inclui uma grande variedade de evidências que vão além do tradicional exame final de lápis e papel. É um sistema de controle de qualidade pelo qual pode ser determinada, em certa etapa do processo ensino-aprendizagem, a efetividade ou não do processo e, em caso negativo, que mudanças precisam ser feitas para assegurar sua efetividade antes que seja tarde" (BLOOM et al., 1971 apud BARBOSA, 2003).

Nas literaturas pesquisadas como, por exemplo, a de Otsuka et al. (2005) referentes à avaliação, observou-se que a mais cogitada e significativa é a avaliação formativa. Esta avaliação busca identificar falhas do sujeito da aprendizagem e em vista disso procurar quais aspectos de instrução devem ser modificados e fornecer o feedback ao professor e ao aprendiz.

No contexto educacional, a avaliação pode ser classificada em diagnóstica, somativa e formativa (BLOOM et al., 1971 apud BARBOSA, 2003).

A avaliação diagnóstica possui o papel de permitir um ajuste recíproco aprendiz/programa de estudos, seja pela modificação do programa, que será adaptado aos aprendizes, seja pela orientação dos aprendizes para subsistemas de formação mais adaptados a seus conhecimentos e competências atuais.

A avaliação somativa ocorre depois da ação de formação; tem a função de verificar se as aquisições visadas pela formação foram feitas.

A avaliação formativa situa-se no centro da ação de formação. Seu papel principal é contribuir para uma boa regulação da atividade de ensino. Trata-se de levantar informações úteis à regulação do processo ensino-aprendizagem; fornece feedback ao aprendiz do que aprendeu e do que precisa aprender e ao professor, identificando as falhas dos aprendizes e quais os aspectos da instrução que devem ser modificados; busca o atendimento às diferenças individuais dos alunos e a prescrição de medidas alternativas de recuperação das falhas de aprendizagem (BLOOM et al., 1971 apud BARBOSA 2003).

4.3.1 Acompanhamento do Aprendiz

Hoje em dia uma das técnicas aplicada no acompanhamento do aprendiz à distância é a avaliação informal. Segundo Silva (2002), o acompanhamento do aprendiz no ambiente de EAD é uma forma de avaliação informal que permite ao professor identificar problemas no processo de aprendizado; constitui-se em um processo contínuo que se inicia no momento em o curso fica disponível aos aprendizes. Portanto, esse acompanhamento é caracterizado pela necessidade do professor obter informações concretas sobre o aprendiz de um curso.

Então, o processo de avaliação no ensino tradicional que os professores utilizam para avaliar/acompanhar seus alunos não se restringe à utilização de métodos formais como, tais como, provas e exercícios. Os professores utilizam, também, métodos informais para perceber o comportamento do aprendiz. Essa percepção é alcançada através de feedback presencial recebido e que permitem avaliar o processo de ensino-aprendizagem. Mas, devido à ausência do contato físico entre professor e aprendiz nos ambientes de ensino à distância essa percepção não é alcançada de uma forma transparente como no ensino tradicional. Deste modo, os

ambientes de ensino-aprendizagem virtuais fornecem meios de avaliar e acompanhar o aprendiz com o objetivo de perceber o comportamento do aprendiz.

Desta maneira, o acompanhamento na educação presencial difere na educação a distância porque aquela é realizada através da observação do professor. Nos ambientes de ensino-aprendizagem virtuais para EAD, a observação é feita por meio de o registro das interações das atividades do aprendiz com o seu ambiente e, utilizando os meios de comunicação que os ambientes virtuais oferecem.

Assim, os ambientes de ensino-aprendizagem buscam meios/formas de capturar a distância algumas características do aprendiz e analisá-las de modo semelhante ao ensino tradicional. Portanto, o grau de interesse, a participação e o comportamento social podem ser vistos pela ótica computacional, considerando, basicamente, as interações do aprendiz com o ambiente de ensino a distância.

Conforme mencionado previamente na definição de EAD no capítulo 1, a EAD é uma forma de ensino que possibilita a auto-aprendizagem, com mediação de recursos didáticos sistematicamente organizados, apresentados em diferentes suportes de informação, utilizados isoladamente ou combinados, e veiculados pelos diversos meios de comunicação. Por isso, necessita de métodos de avaliação e acompanhamento do aprendiz como, por exemplo, técnicas de observação, participação nos debates, nos seminários e em outras técnicas de aprendizagem.

Portanto, na EAD a avaliação tem como propósito de medir a qualidade de processos abrangentes através de critérios de avaliação. Então, pode-se concluir que a avaliação em EAD deve ser constante e professores e aprendizes precisam adotar uma atitude favorável à avaliação e auto-avaliação permanentes como princípio fundamental da educação.

Segundo Otsuka et al. (2005) e Kemczinski (2005), nos mecanismos de acompanhamento do aprendiz disponíveis na maioria dos ambientes virtuais há conjunto de informações pertinentes de como acompanhar e monitorar os aprendizes. Esses mecanismos apresentam os resultados gerados em forma de tabelas e/ou gráficos.

Desta forma, há uma sobrecarga de informações, pois o professor analisa esses resultados gerados pelos mecanismos manualmente ocasionando, portanto, um acúmulo de resultados a serem analisados. Em visto disso, esses mecanismos não são suficientes para suporte efetivo no acompanhamento do aprendiz.

Baseado nessa necessidade de analisar um volume muito grande de informações é que pesquisas atuais se concentram em como ajudar o professor a lidar com o enorme volume de informações geradas pelas atividades dos aprendizes de um curso a distância.

5 INTELIGÊNCIA ARTIFICIAL

No princípio, os computadores foram desenvolvidos para realizar tarefas bem estruturadas e formalizadas, tais como: cálculos numéricos e manipulação de grandes volumes de dados. Para a execução dessas tarefas as máquinas apresentam um desempenho muito superior em relação ao ser humano. Mas, quando os computadores foram utilizados para realizar tarefas de natureza simbólica como, por exemplo, a compreensão de linguagens naturais e imagens, o planejamento e a resolução de problemas, o desempenho dos computadores foi muito baixo devido à complexidade para a compreensão dessas tarefas e a formulação de uma solução computacional. Entretanto, essas tarefas para os seres humanos são de resolução natural. Segundo, Brookshear (2000, pág. 373), “Máquinas algorítmicas são projetadas para executarem tarefas precisamente definidas, com velocidade e exatidão, e fazem isso extremamente bem. Todavia, máquinas são desprovidas de bom senso”.

Em meados de 1950 os pesquisadores, Herbert Simon, Allen Newell e Cliff Shaw conduziram as primeiras experiências em codificação de programas computacionais objetivando simular o processo do pensamento humano. Como consequência dessa experiência resultou no programa chamado Teoria da Lógica, no qual consiste de regras que prova axiomas, capaz de resolver rapidamente trinta e oito dentre cinquenta e dois problemas. Isso foi o maior passo no desenvolvimento da Inteligência Artificial naquela época (KRISHNAMOORTHY et al., 1996).

A partir do momento em que o homem percebeu que era viável modelar a mente humana, usando técnicas e metodologias de programação, deu-se origem à Inteligência Artificial (IA). Pode-se definir IA como o ramo da Ciência da Computação que almeja projetar sistemas computacionais inteligentes. O termo Inteligência Artificial nasceu no ano de 1956 na primeira

conferência organizada por John McCarthy, Marvin Minsky, Nathaniel Rochester e Claude Shanon no Colégio Dartmouth. Nesta conferência o pesquisador John McCarthy, o desenvolvedor da linguagem de programação LISP, propôs o termo Inteligência Artificial.

Portanto, estudos na área de IA, que iniciou no final dos anos 50, objetiva desenvolverem metodologias computacionais de processos peculiares à cognição humana, tais como: raciocínio, comunicação em linguagem natural e aprendizagem. São várias as definições e diversos autores abordam o tema IA. Apresenta-se a seguir uma definição dos seguintes autores, Chaniark et al. (1985). “Inteligência artificial é o estudo das faculdades mentais através do uso de modelos computacionais”.

Para simular a capacidade humana e para que a simulação do funcionamento da mente possa ser implementada através da máquina é necessário, primeiramente, responder algumas questões do tipo:

- Como ocorre o pensar?
- Como o homem extrai conhecimentos do mundo?
- Como a memória, os sentidos e a linguagem ajudam no desenvolvimento da inteligência?
- Como surgem as idéias?
- Como a mente humana processa as informações e infere conclusões?
- etc.

Uma das linhas de pesquisa da IA é a Inteligência Artificial Distribuída (IAD) que objetiva compreender e modelar ações e conhecimento

nos quais vários sistemas interagem para o alcance de um objetivo. Duas áreas diferenciadas fundamentais da IAD são: Resolução Distribuída de Problemas e Sistemas Multiagente (SMA). A resolução distribuída de problemas tem como objetivo solucionar um determinado problema através da divisão de tarefas em módulos, esses cooperam entre si para atingir sua meta, ou seja, os resultados com uma coordenação centralizada para compor um resultado global. Por outro lado, SMAs estão interessados no comportamento individual e coletivo de unidades computacionais de software, denominados agentes inteligentes, que resolvem problemas, sem a necessidade de uma coordenação centralizada.

Desenvolver uma máquina com características semelhante ao homem, sempre foi um grande fascínio. Atualmente, existem alguns sistemas capazes de simplificar, melhorar e aumentar a produtividade em tarefas de natureza simbólica, outrora, a resolução dessas tarefas era exclusiva a seres humanos.

Para tanto, faz-se necessário a definição e compreensão de agentes computacionais, que são módulos de softwares capazes de realizar tarefas de natureza simbólica. Na próxima seção apresenta-se uma abordagem mais detalhada sobre a tecnologia de agentes.

5.1 Agentes

A tecnologia de desenvolvimento de agentes não é relativamente nova, ainda

mesmo por intermédio de executores conforme o esquema da figura 2. Observa-se que um agente tem uma memória interna a qual é atualizada com novas informações através de suas percepções do seu ambiente. Essa memória é utilizada nos procedimentos de tomada de decisão que geram ações. Essas ações realizadas são mantidas como histórico na memória.

Figura 2: As interações de um agente genérico com seu ambiente.
Fonte: RUSSEL et al., 1995.

Segundo Placca (1999), “Agentes computacionais são tipos especiais de programas de computador que atuam em nome de outrem (agentes humanos ou computacionais), dotados de autonomia para executar tarefas e realizar operações, e que buscam atingir o objetivo especificado pelo seu mandante”.

Portanto, nota-se que a autonomia é a característica principal de um agente. Este deve tomar decisões e ações importantes para a realização de um objetivo ou tarefa (ne que 4.7o6)]TJ190.00TjT3emaingir o ntos04a

Além da autonomia, existem algumas propriedades desejáveis para que o agente se torne flexível o suficiente para alcançar seus objetivos, a saber:

a) Reatividade: É a capacidade de reagir rapidamente a alterações no ambiente, ou seja, percebe as modificações no meio e responde de modo oportuno;

b) Pró-atividade: É a capacidade que o agente possui para tomar iniciativa. Eles não respondem simplesmente de acordo com o meio, ou seja, possuem a capacidade de exibir os seus comportamentos baseados em objetivos;

c) Habilidade social: Os agentes são capazes de trocar informações com ambiente, com os outros agentes e, possivelmente, com humanos, para alcançar seus objetivos. Para tanto, os agentes usam as linguagens e os protocolos padrões para cooperar e colaborar para alcançar objetivos comuns;

Os agentes podem ser classificados em dois tipos fundamentais: reativos e cognitivos. Basicamente, as arquiteturas desses agentes diferem uma das outras pelo mecanismo utilizado para decidir a ação a partir de uma percepção (ALVES, 2005).

Os agentes cognitivos são aptos para examinar e dialogar antes de resolver qualquer tipo de problema, ou seja, são capazes de raciocinar a respeito de suas intenções, que são representadas internamente. Sua representação interna e seus mecanismos de inferências permitem atuar independentemente dos outros agentes e lhe dão uma grande flexibilidade na forma de expressão de seu comportamento. Devido a sua capacidade de raciocínio baseado nas representações do mundo, são capazes ao mesmo tempo de memorizar situações, analisá-las e prever possíveis reações para as suas ações. Estes agentes são dotados de conhecimentos, competências,

intenções e crenças, que são subsídios importantes para que eles possam coordenar as suas ações visando a resolução de um problema.

Portanto, para resolver um problema, o agente cognitivo deve possuir conhecimento sobre o problema a ser solucionado. Este conhecimento inserido no agente pode ser adquirido através da interação com outros agentes, ou através da percepção de mudanças no ambiente computacional. Os objetivos que devem ser executados pelos agentes, são alcançados através da interação com os outros agentes, ou através da observação de mudanças no ambiente. A figura 3 apresenta um modelo genérico de agente cognitivo.

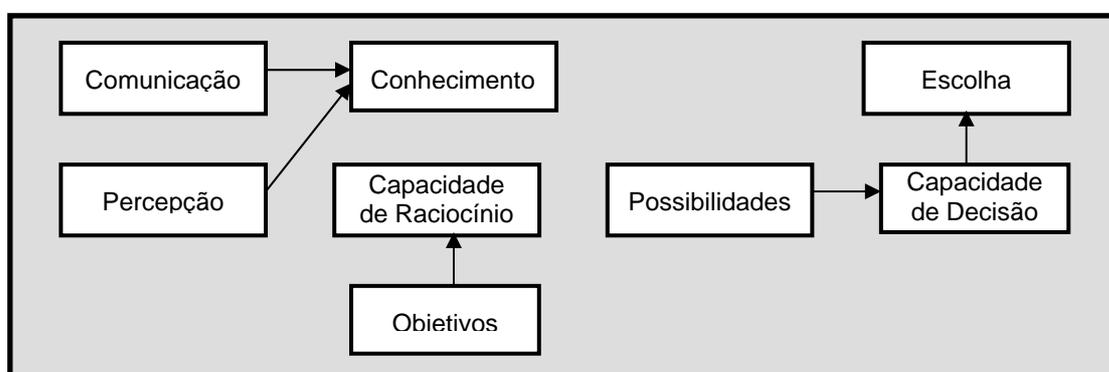


Figura 3: Modelo de agente cognitivo.

Quando é definido um objetivo, o agente usará sua capacidade de raciocínio para planejar como melhor executá-lo. O agente pode pensar em diversas soluções, ou possibilidades, para que este objetivo seja alcançado. Deste modo, ele deve realizar a escolha da melhor solução usando sua capacidade de decisão.

Agentes reativos: os agentes reativos utilizam uma arquitetura de agentes baseada em estímulo-resposta e não possuem um histórico de suas ações passadas, nem fazem previsão de atos futuros. Não tem representação do seu ambiente ou de outros agentes e são incapazes de prever e antecipar ações. Segundo Marietta (2000), o comportamento do sistema é explicado pela existência da inteligência emergente, pois não pode ser explicado pela arquitetura dos agentes (que é elementar), nem pelas estruturas de

comunicação e cooperação (que são baseadas em estímulo-resposta). Assim, o foco de um agente reativo vem da capacidade de formar um grupo e construir colônias capazes de adaptar-se a um ambiente. Um exemplo é o caso da abelha, um único componente não apresenta muita inteligência, mas quando age em grupo comporta-se como uma entidade única, permitindo distinguir padrões de comportamento social.

5.2 Sistema MultiAgente

Um sistema multiagente (SMA) é composto por vários agentes que atuam ou trabalham em conjunto de forma a realizar tarefas ou alcançar seus objetivos que estão além das suas habilidades individuais. Os agentes dentro de um sistema multiagente podem ser heterogêneos ou homogêneos, colaborativos ou competitivos, ou seja, a definição dos tipos de agentes depende da finalidade da aplicação que o sistema multiagente está inserido.

Segundo Hübner (1995), SMA é uma abordagem para solucionar problemas específicos dividindo o trabalho entre muitos agentes que cooperam interagindo e trocando conhecimentos sobre o problema e sua solução, sendo que estes agentes são dotados de certa autonomia e inteligência.

Os Sistemas Multiagentes preocupam-se com a formação do agente. Portanto, o projetista de SMA modela, primeiramente, os agentes para depois analisar em que área essa sociedade de agentes irá atuar. Os agentes são capazes de solucionar diversos tipos de problema.

O esquema de resolução de problemas na abordagem SMA pode ser observado na figura 4.

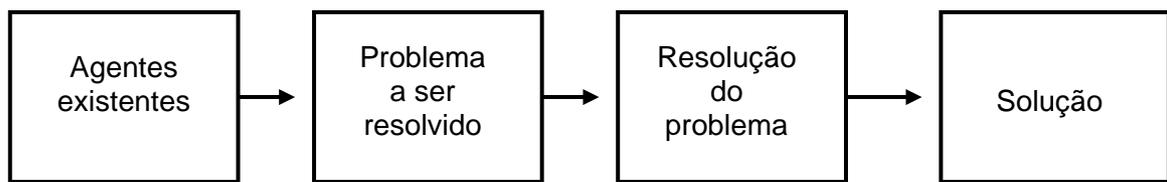


Figura 4: Fluxo do projeto de um SMA
Fonte: Sichman et al., 1992.

Os autores Demazeau (1995) e Sichman et al. (1997) destacam algumas considerações importantes para a abordagem SMA, que são:

- Os agentes devem ser capazes de decompor as tarefas baseados no conhecimento que eles possuem de si próprios e dos outros agentes.
- Como os agentes são autônomos, eles podem possuir objetivos próprios e decidir o quê fazer, a qualquer momento.
- Os agentes possuem capacidades para resolver seus problemas e os problemas que surgirem no ambiente.
- Os agentes podem entrar e sair do ambiente a qualquer momento. Portanto, em um SMA os agentes devem ser capazes de modificar o conhecimento que possuem dos outros agentes do ambiente.
- Os agentes devem ser capazes de reconhecer as modificações no ambiente quando estas ocorrerem, alterando sua representação interna no ambiente.

Pode-se observar que os agentes para participar de um SMA têm que possuir algumas capacidades específicas, tais como: comunicação, interação e coordenação.

5.3 Comunicação entre Agentes

A Comunicação é um processo em que a informação é compartilhada numa transação entre agentes. Tem como principal objetivo o compartilhamento do conhecimento e a coordenação de atividades entre estes agentes. Assim, padronizaram a forma de comunicação e a forma de acessar os serviços comuns entre os diversos agentes.

Observa-se que os agentes possuem determinados objetivos para serem alcançados. Então, faz-se necessário a cooperação entre estes agentes. Para que essa cooperação aconteça, é fundamental que seja bem definida uma estrutura que possibilite a interação entre eles. Nestas interações irão ocorrer trocas de conhecimentos, objetivos, planos ou escolhas através de comunicação, que pode ocorrer de duas formas, que são:

Comunicação indireta – os agentes não se conhecem e, desta maneira, a comunicação irá ocorrer através de uma estrutura de dados compartilhada. A figura 5 apresenta um modelo de interação de forma indireta entre agentes.

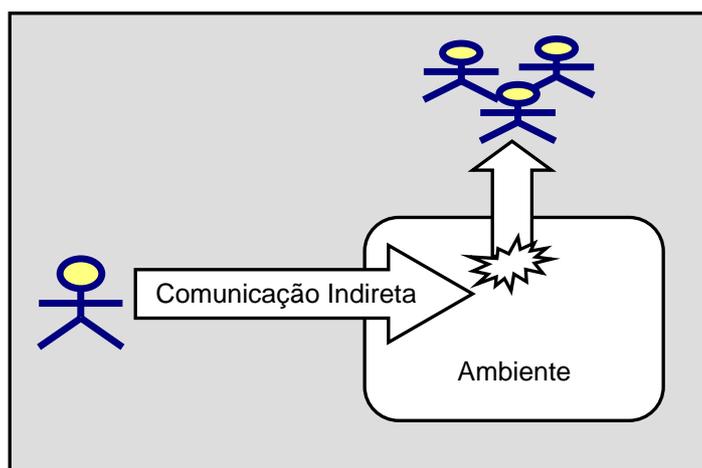


Figura 5: Modelo de interação indireta

Comunicação direta – os agentes se conhecem e, por isso, trocam informações diretamente entre si. A figura 6 apresenta um modelo de interação de forma direta entre agentes.



Figura 6: Interação entre agentes de forma direta

Nas duas seções seguintes, 5.3.1 e 5.3.2, será feita uma abordagem mais detalhada sobre as formas de comunicação entre agentes.

5.3.1 Comunicação Indireta

Em um SMA, quando os agentes não se conhecem, utiliza-se uma estrutura de dados compartilhada, para que os agentes possam interagir entre si. Neste tipo de sociedade, uma abordagem a ser utilizada é a arquitetura de quadro negro (blackboard).

Nesta arquitetura, os agentes não se comunicam entre si diretamente, mas sim através de um quadro-negro no qual existe uma divisão em regiões ou níveis visando facilitar a busca de informações, sendo uma espécie de repositório que fornece um meio de interação entre os agentes. Quando os agentes necessitam de alguma informação, eles escrevem seu pedido no quadro à espera que outros agentes respondam à medida que acessem o mesmo. Este modelo de comunicação foi proposto na década de 1970 para um sistema de interpretação da fala chamado Hearsay (OLIVEIRA, 1996 apud JAQUES, 1999). O sistema, Hearsay, inicialmente possui um sistema de controle que supervisiona a escrita e a leitura no quadro negro. A figura 7 mostra o esquema da arquitetura do quadro negro.

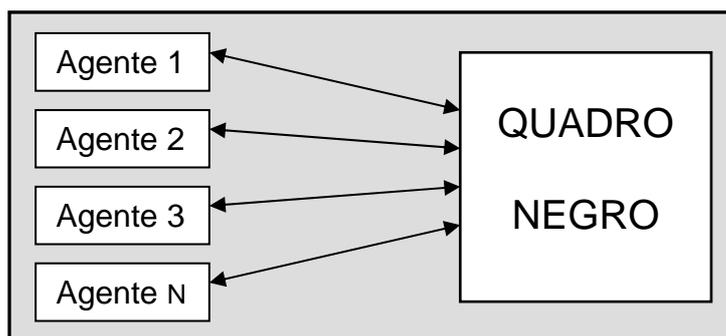


Figura 7: Estrutura de Quadro-negro.

Esta arquitetura possui uma estrutura única e compartilhada entre os diversos agentes, na qual as informações serão escritas e lidas durante o desenvolvimento das tarefas. A comunicação nunca ocorre diretamente entre os agentes, assim, eles devem consultar a estrutura de tempos em tempos para verificar se existe alguma informação destinada a eles.

Apesar de ser simples e eficiente na sua implementação, um dos problemas que ocorrem na arquitetura de quadro negro é a sincronização e a resolução de conflitos entre os agentes por causa da centralização e o controle dessas informações.

5.3.2 Comunicação Direta

Na sociedade em que os agentes se conhecem, a comunicação pode ser realizada de forma direta, ou seja, os agentes trocam mensagem diretamente entre si. Para isso, estes agentes devem ser conhecidos no sistema, assim, cada agente possuirá uma representação dos outros agentes, tais como: capacidades, objetivos, conhecimentos e crenças.

Na arquitetura de troca de mensagens, os agentes comunicam-se, uns com os outros, diretamente através de mensagens assíncronas. As trocas de mensagens são potencialmente mais ricas na interatividade do que em arquiteturas do tipo quadro negro. Porém, a implementação torna-se mais complexa, devido à necessidade de serem utilizados protocolos e linguagens

que definem os métodos de diálogo que serão realizados pelos agentes, para os diversos tipos de interação no sistema. De acordo com a necessidade, cada sistema utilizará um protocolo de comunicação mais apropriado para a aplicação e a tecnologia utilizada.

Segundo Alves (2005, pág. 25) quando o número de agentes de um SMA é muito grande a emissão de uma mensagem em broadcasting¹ pode ocasionar uma demora inviabilizando todo processo de comunicação do sistema. Visando remediar esse contratempo, surgiu a arquitetura federativa que divide o SMA em grupos ou federações de agentes, de acordo com o critério de agrupamento escolhido. Nestas federações há um agente denominado de facilitador, que é responsável por receber a mensagem que chega a cada federação e encaminhá-la para o agente destinatário de uma federação. Portanto, com esse tipo de arquitetura permite a diminuição do fluxo de mensagens desnecessárias entre os demais agentes que constituem a sociedade, porque os facilitadores têm a capacidade de remetê-las ao respectivo destinatário sem a necessidade de enviá-las a todos os agentes.

Atualmente, existem dois padrões de comunicação direta baseados em trocas de mensagens, que são: a KQML (Knowledge Query and Manipulation Language) e a FIPA-ACL (Agent Communication Language).

KQML é uma linguagem e protocolo de comunicação de alto nível de consultas, manutenção, ações e serviços de bases de conhecimento. Ela é resultante de um projeto formado pelo DARPA (Defense advanced Research Projects Agency) e por outras agências ligadas ao departamento de defesa norte-americano. É uma linguagem que é utilizada para compartilhar informações e conhecimentos entre os agentes, e caracteriza-se tanto por uma formatação das mensagens como por um protocolo para manipular estas mensagens que suportam em tempo de execução, o compartilhamento de conhecimento e informação. O objetivo é desenvolver metodologias e técnicas

¹ Envio simultâneo de uma mensagem para múltiplos destinatários.

para a construção de bases de conhecimento, compartilháveis entre aplicações.

As mensagens contêm os parâmetros de transporte com os endereços do agente transmissor e receptor envolvidos na comunicação. Estes parâmetros podem ser combinados com um serviço de nomes, endereçamento e registro para a conversão em endereços compatíveis com a camada de transmissão.

A linguagem KQML oferece diversos tipos de mensagens, conhecidas como performativas, que representam o objetivo do agente ao enviar uma mensagem. São mensagens predefinidas (intenção da mensagem), que definem operações possíveis de serem executadas entre os agentes. Esta linguagem possui 36 performativas que são visualizadas na tabela 1.

Tabela 1: Quadro de representação das Performativas KQML

PERFORMATIVAS	IDENTIFICADORES
Consultas Básicas	Ask, Ask-if, Ask-one, Ask-all
Troca de informação	Tell, Untell, Insert, Uninsert, Delete-one, Delete-all, Undelete
Requisição de Serviço	Subscribe, Achieve, Unachieve, Broker-one, Broker-all
Sincronismo do envio de mensagens	Error, Sorry, Standby, Ready, Next, Forward, Broadcast, Rest, Discard, Transport-address
Gerenciamento de agentes	Register, Unregister, Advertive, Unadvertive, Recomend-all, Recruit-one, Recruit-all

Também possui um conjunto de parâmetros chaves reservados, sendo útil para estabelecer grau de uniformidade dos parâmetros e suporte para que os agentes entendam performativas desconhecidas, mas com parâmetros conhecidos. Os parâmetros reservados e seus significados são vistos na tabela 2.

Tabela 2: Quadro dos parâmetros reservados de KQML.

PARÂMETROS CHAVE	SIGNIFICADO
:content	A informação sobre qual performativa expressa uma atitude
:force	Se o agente transmissor irá sempre negar o significado da performativa
:in-reply-to	O rótulo esperado em resposta
:language	O nome de uma linguagem contida no parâmetro :content
:ontology	O nome da ontologia usado no parâmetro :content
:receiver	O agente receptor atual da performativa
:reply-with	Se o agente transmissor espera uma resposta, e se sim, um rótulo para a resposta
:sender	O atual agente transmissor da performativa

A linguagem KQML pode ser dividida em três camadas, que são:

Camada de conteúdo: Nesta camada fica embutido o conteúdo da mensagem, representando por uma expressão em alguma linguagem;

Camada de Comunicação: Nesta camada são codificados os parâmetros de alto nível (identidade do agente emissor e receptor, identificação da mensagem);

Camada da mensagem: Esta camada identifica o protocolo usado para transmitir a mensagem e a performativa que o agente transmissor anexa ao conteúdo.

A FIPA-ACL é uma linguagem de comunicação de agentes desenvolvida pela FIPA (Foundation Intelligent Physical Agents), uma organização sem fins lucrativos fundada em 1996 em Genebra, Suíça, que tem por objetivo estabelecer padrões para o desenvolvimento de agentes inteligentes. Sua sintaxe é semelhante ao KQML, embora o conjunto de performativas (atos comunicativos) seja diferente, sua especificação consiste de um conjunto de tipos de mensagens e descrições dos efeitos da mensagem sobre os agentes transmissores e receptores. Os parâmetros nativos da FIPA-

ACL são divididos em cinco categorias, que são: tipo do ato comunicativo, participantes da comunicação, conteúdo de mensagem, descrição do conteúdo e o controle da conversação. Na tabela 3, são mostrados os parâmetros e suas respectivas categorias.

Tabela 3: Parâmetros das mensagens FIPA-ACL

TIPO DO ATO COMUNICATIVO	
Performative	Denota o tipo de ato comunicativo da mensagem.
Participantes da Comunicação	
Sender	É o agente emissor da mensagem.
Receiver	É o agente receptor da mensagem.
Reply-to	Indica que as próximas mensagens dessa conversação deverão ser enviadas para o agente indicado pelo parâmetro reply-to.
CONTEÚDO DE MENSAGEM	
Content	É o conteúdo ou conhecimento transportado pela mensagem.
DESCRIÇÃO DO CONTEÚDO	
Language	É a linguagem na qual o conhecimento está expresso.
Encoding	Aponta a codificação utilizada na expressão da linguagem de conteúdo.
Ontology	É a ontologia utilizada para dar significado à expressão de conteúdo.
CONTROLE DE CONVERSAÇÃO	
Protocol	É o protocolo de interação utilizado, para essa mensagem, pelo agente emissor.
Conversation-id	Introduz uma expressão que será usada para identificar a conversação em andamento.
Reply-with	Introduz uma expressão que será usada pelo agente que responderá a essa mensagem para identificá-la.
In-reply-to	É a expressão que referencia a mensagem à qual se está respondendo.
Reply-by	Explicita um tempo máximo, durante o qual o agente emissor estará esperando por uma resposta a esta mensagem.

O conteúdo de uma mensagem, indicado pelo parâmetro content, referencia a informação sobre 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. Cada mensagem FIPA-ACL transporta um ato comunicativo, que representa a vontade de um agente sobre determinada informação, carregada pela mensagem. A tabela 4 mostra os atos comunicativos de padrões da linguagem FIPA-ACL.

Tabela 4: Atos Comunicativos de FIPA-ACL.

Ato comunicativo	Resumo
Accept Proposal	Informa a aceitação de uma proposta prévia para a execução de uma determinada ação. O conteúdo da mensagem deve conter a ação a ser feita e a condição aceita.
Agree	Informa a concordância em executar alguma ação, possivelmente no futuro. O conteúdo da mensagem deve conter a ação (futura) e a condição aceita.
Cancel	Informar para um determinado agente que ele não necessita mais executar a ação pedida anteriormente. No conteúdo está expressa a ação que não é mais requerida.
Cfp	O ato <i>cfp</i> (<i>Call for Proposal</i>) solicita propostas para a execução de uma determinada ação. A mensagem deve conter qual a ação deve ser feita e qual a pré-condição para esta ação.
Confirm	O agente emissor informa ao agente receptor que uma dada proposição é verdadeira, se o agente receptor estava (reconhecidamente) incerto disso. O conteúdo é a proposição.
Disconfirm	O agente emissor informa ao agente receptor que uma dada proposição é falsa, se o agente receptor estava (reconhecidamente) certo de que ela era verdadeira. O conteúdo da mensagem é a proposição.
Failure	O agente emissor informa ao receptor que tentou fazer uma ação e que essa tentativa falhou. O conteúdo é composto da ação que falhou e da razão da falha.
Inform	O agente emissor informa ao receptor que uma dada proposição é verdadeira. O conteúdo da mensagem é a própria posição.
Inform-If	É um ato composto que serve para o agente emissor informar ao agente receptor se uma dada proposição é verdadeira ou não. O conteúdo da mensagem é a própria posição.
Inform-Ref	É um ato composto que serve para o agente emissor informar ao agente receptor o objeto que corresponde a um dado descritor. O conteúdo da mensagem é uma expressão referencial, um descritor de objeto.
Not Understood	O agente emissor informa ao agente receptor que não entendeu uma ação ou ato prévio do agente receptor. O conteúdo da mensagem é composto do ato ou ação não compreendida e de uma explicação do que não foi compreendido.
Propagate	Serve para que o agente emissor solicite a manipulação da mensagem encapsulada em anexo como se tivesse sido emitida diretamente por ele, mas que também busque outros agentes que se encaixam num descritor, também passado em anexo, e reenvie a mensagem para os agentes selecionados pela busca. O conteúdo da mensagem é composto de dois elementos: um descritor dos outros agentes que deverão receber a mensagem sendo propagada e um ato comunicativo completo, contendo a mensagem encapsulada.
Propose	Serve para que agente emissor envie ao receptor uma proposta para efetuar alguma ação, dadas certas pré-condições. O conteúdo da mensagem é composto da descrição da ação sendo proposta e da pré-condição na execução dela.
Proxy	O agente emissor quer que o agente receptor busque outros agentes, que se encaixem na descrição passada em anexo, e envie a mensagem em anexo para esses agentes. O conteúdo da mensagem é composto de dois elementos: um descritor dos outros agentes que deverão receber a mensagem sendo passada por procuração e um ato comunicativo completo, contendo a mensagem encapsulada.
Query-If	Representa a ação de perguntar a um agente se uma determinada proposição é verdadeira ou não. O conteúdo da mensagem é a própria posição.

Query-Ref	Representa a ação de perguntar a um agente Qual o objeto que atende uma determinada expressão referencial. O conteúdo da mensagem é a própria expressão referencial (um descritor do objeto).
Refuse	Representa a ação de se recusar a executar uma dada ação e explicar a razão porque. O conteúdo é composto da ação recusada e da explicação da recusa.
Reject-Proposal	Representa a ação de rejeitar a execução de alguma ação, durante uma negociação. O conteúdo é composto da ação rejeitada e da explicação do porquê da rejeição.
Request	O agente emissor solicita ao receptor que ele execute alguma ação (possivelmente um outro ato comunicativo). O conteúdo da mensagem é uma ação a ser feita.
Request-When	O agente emissor solicita ao receptor que ele execute alguma ação, quando uma dada proposição for verdadeira. O conteúdo da mensagem é composto da ação a ser feita e da proposição.
Request-Whenever	O agente emissor solicita ao receptor que ele execute alguma ação, assim que uma dada proposição for verdadeira e que a continue executando cada vez que ela se tornar verdadeira novamente. O conteúdo da mensagem é composto da ação a ser feita e da proposição.
Subscribe	Solicita a notificação do valor das atualizações no valor de uma dada referência. O conteúdo da mensagem é composto de uma expressão referencial (uma descrição do valor a ser notificado).

Na figura 8, observa-se uma estrutura de uma mensagem com as partes que compõem uma mensagem básica.

```
(communicative
  :sender <valor>
  :receiver <valor>
  :sender <valor>
  :content <valor>
  :ontology <valor>
  :conversation-id <valor>
  ...)
```

Figura 8: formato de uma mensagem FIPA-ACL

5.4 Plataforma de Agentes

Atualmente, existem Plataformas de Agentes como, por exemplo, ZEUS, FIPA-OS e JADE que auxiliam o desenvolvimento de Sistemas Multiagentes, facilitando, portanto, todo o processo de implementação fornecendo infraestrutura requerida. As plataformas ZEUS e JADE utilizam o padrão FIPA. No ambiente NetClass, utiliza-se o Framework JADE, por este fornecer bibliotecas prontas, ferramentas de gerenciamento, ser implementado totalmente em

JAVA, por ser consolidada e aceita nas comunidades científicas, estável e constante aperfeiçoamento.

5.4.1 JADE (Java Agent DEvelopment Framework)

Java Agent DEvelopment Framework (JADE) é um framework de software para o desenvolvimento e execução de sistemas multiagentes. Foi desenvolvido pelo CSELT (Centro de Estudo e Laboratório de Telecomunicação), atualmente conhecido como Tilab (Telecomm Italia Lab), juntamente com o Grupo de Engenharia de Computação da Universidade de Parma.

O Framework JADE é implementado completamente na linguagem Java. JADE simplifica a implementação de Sistemas multiagentes através de um middleware, que cumpre com as especificações da FIPA (Foundation for Intelligent Physical Agents), e, também, com um conjunto de ferramentas gráficas que suporta a depuração e a fase de desenvolvimento. JADE é um software open source sob os termos da licença LGPL (Lesser General Public License). A última versão do JADE é 3.4.1, que foi disponível em 17 de novembro de 2006.

O termo middleware é tencionado para descrever todas as bibliotecas de alto nível que deixa aplicação mais efetiva e fácil de desenvolver por fornecer serviços genéricos úteis não apenas por aplicação simples, mas, preferencialmente, para uma variedade de aplicações, por exemplo, comunicação, acesso a dados, codificação e controle de recursos.

O padrão FIPA especifica o modelo padrão (referência) que uma plataforma de agente e um conjunto de serviços devem ser fornecidos. A coleção destes serviços, e suas interfaces, representam as regras normativas que permitem a sociedade de agente existir, funcionar, e ser gerenciada.

A plataforma JADE inclui o sistema gerenciador de agentes (AMS – Agent Management System), o diretório facilitador (DF – Directory Facilitator) e o canal de comunicação de agentes (ACC – Agent Communication Channel) de acordo com a padronização FIPA. Esses três componentes são automaticamente iniciados quando a plataforma é instanciada.

O AMS é responsável pelo serviço de registro e o cancelamento automático de agente em conformidade aos padrões FIPA, ou seja, é responsável pelo serviço de nomeação que assegura cada agente tem um nome único no ambiente JADE. Quando o agente é instanciado (criado) este obtém seu GUID (Globally Unique Identifier) da plataforma. O DF prover o serviço de páginas amarelas no qual agente pode procurar outros agentes que realizam determinados serviços fornecendo os serviços desejados para realizar seus objetivos. Portanto, com o AMS e DF facilita o trabalho do desenvolvedor, pois, evita que este esteja envolvido com atividades mais complexas (baixo nível).

A plataforma JADE inclui as bibliotecas de classes que programadores utilizam as diretamente e/ou por especialização no desenvolvimento de agentes e o ambiente de tempo de execução o lugar em que os agentes “moram” e necessita ser ativo em um determinado host para depois os agentes serem executados.

Então, JADE fornece os serviços básicos necessário a aplicações ponto-a-ponto e permite que cada agente descubra dinamicamente outros agentes e se comunicam.

O framework JADE simplifica, portanto, o desenvolvimento de agentes fornecendo serviços que trata da comunicação, do ciclo de vida, do monitoramento da execução, entre outros serviços. Desta forma, a plataforma JADE fornece serviços para garantir um padrão de interoperabilidade entre

sistemas multiagentes, os quais possibilitam e facilitam a comunicação entre agentes.

Portanto, o framework JADE é um middleware, ou seja, uma camada intermediária de desenvolvimento, que pode ser utilizado como referência para implementação de modelos de mais alto nível.

5.4.2 Modelo Padrão da Plataforma JADE

O modelo de arquitetura padrão da plataforma JADE foi especificado pela FIPA, como mostra a figura 9.

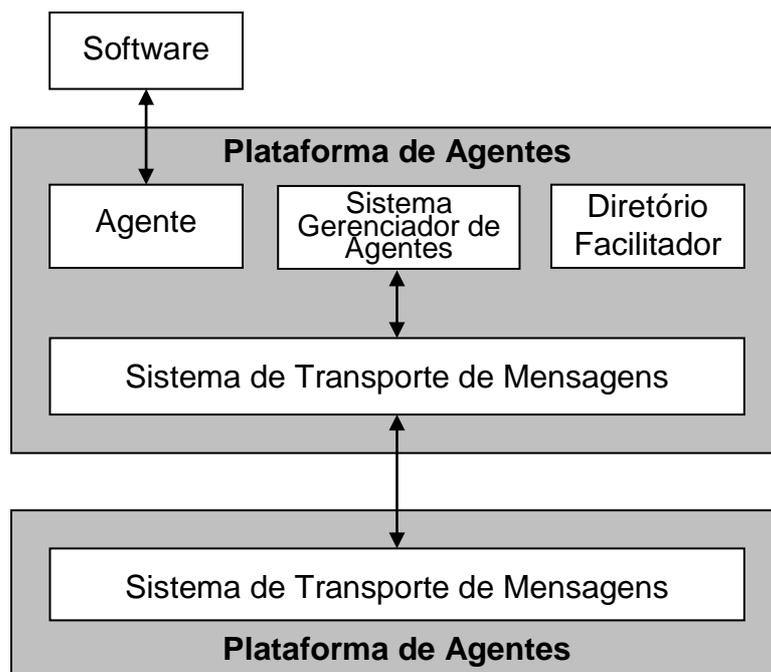


Figura 9: Modelo padrão de plataforma de agentes definido pela FIPA

Fonte: BELLIFEMINE, 2006.

A FIPA identifica os papéis de alguns agentes necessários para a administração da plataforma, que são:

O agente – As suas atividades são definidas de acordo com o objetivo da aplicação. É ele que realiza a comunicação com os agentes através de troca de mensagens e relacionando-se com a aplicação externa.

O sistema gerenciador de agentes (AMS) – Supervisiona o acesso e o uso da plataforma de agentes, parte superior central da figura 9, e apenas um AMS irá existir em uma plataforma. Ele provê guia de endereços (páginas em branco) e controle do ciclo de vida, mantendo um diretório de identificadores de agentes (Agent Identifier – AID) e os estados dos agentes e é o responsável pela autenticação dos 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) – Disponibiliza o serviço de páginas amarelas (yellow pages – YP) dentro da plataforma.

O sistema de transporte de mensagens – É conhecido também como canal de comunicação dos agentes (Agent Communication Channel – ACC), na parte inferior da figura 9, é o responsável por prover toda a comunicação entre os agentes internamente e externamente à plataforma. Todos os agentes, inclusive o AMS e o DF, utilizam esse canal para a comunicação.

Portanto, JADE utiliza esse modelo de plataforma especificado pela FIPA. Na instanciação da plataforma JADE, o AMS e o DF são criados e o ACC é configurado para permitir a comunicação através de mensagens.

5.4.3 Arquitetura Distribuída

Cada instancia em execução de o ambiente de tempo de execução JADE é chamado de um container que pode conter vários agentes. Um conjunto de containers instanciados é denominado de Plataforma. Em uma plataforma deve existir obrigatoriamente somente um container principal sempre ativo.

Nota-se que o container principal é o primeiro a ser inicializado (instanciado) e assegura os agentes especiais, o AMS e o DF, assim que a mesma é carregada, sendo responsável pelos registros de outros containers na plataforma. A plataforma fornece uma camada homogênea que oculta aos agentes a complexidade e a diversidade da base (hardware, sistemas operacionais, tipos de redes, Java Virtual Machine – JVM).

Se em uma mesma rede existirem dois containers principais então isso implica em duas plataformas distintas que podem ter outros containers secundários. No cenário representado na figura 10 tem duas plataformas JADE, plataforma 1 e plataforma 2, contendo 3 e 1 containers respectivamente.

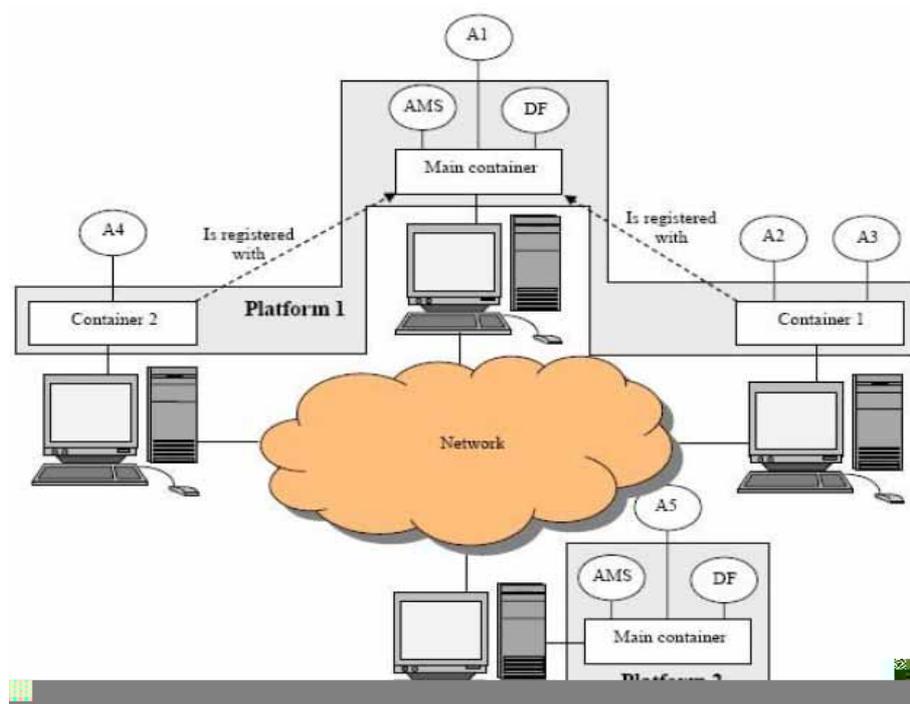


Figura 10: Container e Plataforma

Um agente JADE é identificado por um nome único, conhece o nome dos outros agentes e pode manter uma comunicação transparente com eles a partir de sua localização. Por exemplo, agentes que estão no mesmo container (A2 e A3), agentes em containers diferentes dentro da mesma plataforma de agentes (A1 e A2) ou agentes em plataformas diferentes (A4 e A5).

Observa-se na figura 11, que a arquitetura da plataforma JADE é baseada na coexistência de vários JVM (Java Virtual Machine) que podem estar distribuídas por diversas máquinas, sendo que em cada JVM existe um container de agentes que fornece um ambiente completo para a execução destes agentes e permite que vários agentes sejam executados concorrentemente no mesmo host.

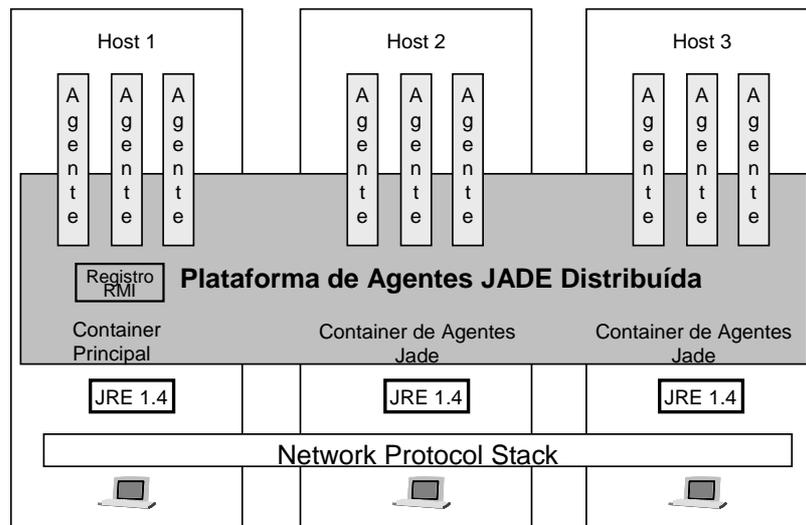


Figura 11: Plataforma de Agente JADE distribuídas em vários containers

Fonte: BELLIFEMINE, 2006

É no container principal que estão localizados o AMS, DF e registro RMI (Invocação de Método Remoto) que java utiliza para registrar e recuperar referências a objetos através de nome, ou seja, um servidor de nome onde fica as referências aos outros containeres de agentes que se conectam na plataforma. Os outros containeres de agentes conectam ao container principal fazendo com que o desenvolvedor fique abstraído da separação física entre os hosts.

5.4.4 Agentes na Plataforma JADE

É importante observar que o padrão FIPA somente estabelece padrões referentes ao comportamento externo de um agente, assim, com a

padronização dos elementos externos adquiri-se a interoperabilidade entre plataformas (FIPA, 2002).

Assim que a plataforma é instanciada, os agentes que estão sendo instanciados recebem seus identificadores únicos globais e o AMS registra-os automaticamente. O nome de cada agente é composto por um apelido mais um símbolo @, seguidos do nome do host.domínio e da porta / JADE, por exemplo: agente@meucomputador:1099/JADE.

Um agente JADE pode ser definido como uma entidade autônoma, que interage com outros agentes através de troca de mensagens a fim de cumprir seus objetivos. A plataforma JADE não especifica o tipo de agente que pode ser desenvolvido utilizando a plataforma.

Um agente definido na plataforma JADE é uma subclasse da superclasse Agent do pacote jade.core.Agent. Esse agente que foi definido herda todos os métodos de sua superclasse. Então, para criar um agente reativo na plataforma JADE é preciso instanciar o objeto da classe jade.core.Agent. O método setup() herdado realiza as inicializações necessárias para que o agente criado desempenhe o seu objetivo. Portanto, em uma visão mais técnica, um agente JADE é simplesmente uma instância da superclasse Agent, na qual os programadores escrevem seus agentes como subclasses e adicionam comportamentos específicos aos objetivos da aplicação e utilizam as capacidades herdadas da classe Agent. Esta provê métodos para executar tarefas básicas, tais como:

- Passagens de mensagens usando objetos ACLMessage.
- Suporte completo ao ciclo de vida dos agentes, incluindo iniciar ou carregar, suspender e matar um agente.
- Escalonamento e execução de múltiplas atividades concorrentes.

- Interação simplificada com sistemas de agentes FIPA para a automação de tarefas comuns de agentes.

O modelo computacional de um agente JADE é multitarefa, ou seja, as tarefas são executadas concorrentemente. Cada tarefa a ser realizada de um agente deve ser codificada como um comportamento. Um escalonador interno da classe superclasse gerencia automaticamente a execução dos comportamentos.

A arquitetura interna de um agente JADE é mostrada na figura 12. Na parte superior da figura estão os comportamentos (Behaviours) ativos dos agentes e representam as tarefas desempenhadas pelo agente. Na parte inferior à esquerda tem a fila privativa de mensagens ACL que todos os agentes têm essa fila onde decide quando ler as mensagens recebidas e quais mensagens ler.

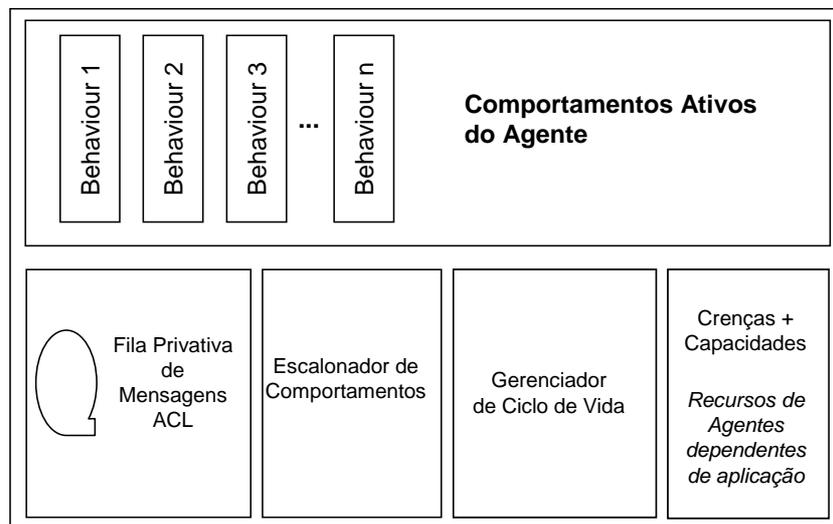


Figura 12: Arquitetura Interna de um Agente JADE

O escalonador de comportamentos e o gerenciador do ciclo de vida no centro da figura. O primeiro é responsável por escalonar a ordem de execução dos comportamentos, se, por exemplo, seguirão a ordem de precedência ou não. O ciclo de vida é o controlador do estado atual do agente.

Os agentes da plataforma JADE podem estar em diversos estados de acordo com o ciclo de vida especificado pela FIPA como mostra a figura 13.

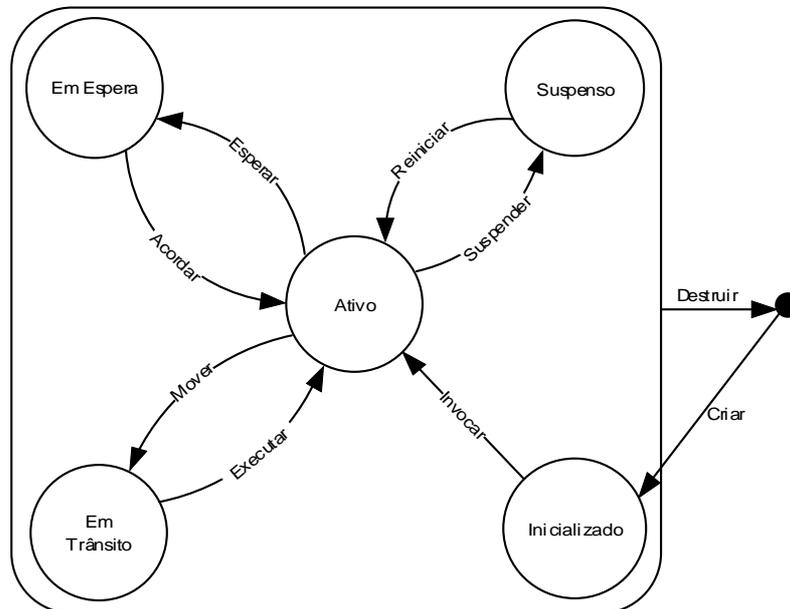


Figura 13: Ciclo de vida de um Agente definido pela FIPA (FIPA 2006)

As transições de estados são feitas por meio de métodos públicos da superclasse Agent como, por exemplo, o método doWait() faz o agente passar do estado ativo para o estado em espera, o método doSuspend() faz a transição do estado ativo ou em espera para o estado suspenso, o método doDelete() faz a transição dos estados ativo, suspenso ou em espera para o estado desconhecido, que causa a destruição do agente, etc. Os estados possíveis para um agente da superclasse Agent são representados por constantes, listadas abaixo:

AP_INIATED: objeto da classe Agent foi instanciado, mas não foi registrado no AMS, ainda não tem AID (nome ou endereço) e por isso não pode se comunicar com outros agentes.

AP_ACTIVE: objeto da classe Agent já foi registrado no AMS, sendo assim, já possui seu AID e já pode ter acesso às funcionalidades da plataforma JADE.

AP_SUSPENDED: objeto da classe Agent está atualmente interrompido. Sua thread interna está suspensa e nenhum comportamento (behaviour) está sendo executado.

AP_WAITING: o objeto da classe Agent está bloqueado. Sua thread está dormindo temporariamente controlada por um monitor Java que irá acordá-lo quando algum evento ocorrer como, por exemplo, a chegada de uma mensagem.

AP_DELETED: objeto da classe Agent está definitivamente encerrado. Sua thread interna foi terminada e o agente não está mais registrado no AMS.

AP_TRANSIT: Um agente móvel entra nesse estado enquanto estiver migrando. O sistema continua armazenando mensagens em um buffer que serão enviadas para a nova localidade.

AP_COPY: Esse estado é usado internamente pelo JADE para agentes que foram clonados.

AP_GONE: Esse estado é usado internamente pelo JADE quando o agente móvel migrou para uma nova localidade e se encontra em um estado estável.

É importante ressaltar que um agente só pode executar seus comportamentos no estado ativo. Também, quando algum comportamento do agente chama o método doWait() ocasionará a interrupção de todas as suas atividades, inclusive, outros comportamentos e não só o comportamento que chamou o método. O método block() da classe Behaviour, ao contrário do método doWait, suspende apenas um comportamento do agente.

Os principais métodos públicos disponíveis na classe Agent, são:

`void setup()`: Método indicado para iniciar a aplicação. Geralmente esse método é sobrescrito, onde os programadores incluem comportamentos necessários ao agente de acordo com a aplicação. É essencial adicionar nesse método algum comportamento, porém o agente já está registrado no AMS e pronto para enviar e receber mensagens.

`void addBehaviour(Behaviour comportamento)`: Adiciona um novo comportamento ao agente, que será executado concorrentemente com outros comportamento. Geralmente usado dentro do método `setup()` para disparar um comportamento inicial.

`void send(ACLMessage mensagem)`: Envia uma mensagem ACL para um determinado agente. O destinatário é especificado no campo `receiver` da mensagem, sendo que vários agentes podem receber a mensagem.

`ACLMessage receive()`: Lê uma mensagem da fila de mensagens do agente. Caso a fila esteja vazia, retorna `null`, caso contrário retorna a primeira mensagem da fila.

5.4.5 Comportamentos

Um agente da plataforma JADE desempenha tarefas e ações através de comportamentos com a finalidade de cumprir os objetivos da sociedade de agentes. Estas tarefas que os agentes desempenham em uma sociedade de agentes são implementadas como comportamentos específicos de um agente.

Desta forma, um agente para realizar um objetivo deve implementar comportamentos específicos, instanciá-los e adicioná-los a si chamando seu método `addBehaviour(Behaviour)` herdado da superclasse `Agent`. As tarefas são codificadas como objetos da classe `Behaviour`. Cada agente na plataforma JADE é representado por uma linha (`thread`) de execução. JADE utiliza uma `thread` por agente ao invés de usar uma `thread` para cada comportamento para

não afetar o desempenho da aplicação, assim, a plataforma mantém uma quantidade pequena de threads em execução.

Os comportamentos adicionados no agente são colocados em uma fila. O escalonador interno à classe Agent, transparente ao programador, gerencia o escalonamento dos comportamentos que estão na fila através de uma política round-robin. Então, com esta política cada comportamento do agente é executado durante uma fatia de tempo, e se este não for completado, volta para o final da fila e espera sua vez de executar novamente, caso haja mais comportamentos na fila.

A classe Behaviour é uma classe abstrata que dispõe de uma estrutura de comportamentos que podem ser implementados. O método principal dessa classe é action() que executa o comportamento. Então, é nesse método que as tarefas realizadas pelo comportamento serão implementadas. O método done() verifica se o comportamento terminou sua execução retornando true ou false. Quando o comportamento termina sua execução ele é retirado da fila de comportamentos do agente. Caso contrário, o método retorna false, deixando a execução do método action() novamente.

Na biblioteca de o JADE existem classes de comportamentos prontas para serem utilizadas. As classes mais gerais são SimpleBehaviour, CompositiveBehaviour e ReceiverBehaviour, sendo que algumas destas são subdivididas em outras classes para tornar suas funcionalidades mais específicas.

- A classe jade.core.behaviours.FSMBehaviour implementa um comportamento composto, baseado no escalonamento por máquina de estados finitos. O FSMBehaviour executa cada comportamento filho de acordo com uma máquina de estados finitos definida pelo usuário. Mais especificamente, cada comportamento-filho representa um estado na máquina de estados finitos. Ela fornece métodos para registrar estado

(sub-comportamentos) e transições que definem como será feito o escalonamento desses comportamentos-filho.

- A classe `jade.core.behaviours.ParallelBehaviour` implementa um comportamento composto por escalonamento concorrente de comportamentos-filho. `ParallelBehaviour` executa seus comportamentos-filho concorrentemente e finaliza quando uma condição particular em seus sub-comportamentos é atingida; por exemplo, quando um número “X” de comportamentos-filho terminarem ou um comportamento-filho qualquer terminar ou quando todos os comportamentos-filho terminarem.
- A classe `jade.core.behaviours.CyclicBehaviour` implementa um comportamento atômico que deve ser executado sempre. Essa classe abstrata pode ser herdada para criação de comportamentos que se manterão executando continuamente.
- A classe `jade.core.behaviours.TickerBehaviour` implementa um comportamento executado periodicamente tarefas específicas. Ou seja, é uma classe abstrata que implementa um comportamento que executa periodicamente um pedaço de código definido pelo usuário em uma determinada frequência de repetições.
- A classe `jade.core.behaviours.OneShotBehaviour` implementa um comportamento atômico que é executado uma única vez. Essa classe abstrata pode ser herdada para a criação de comportamentos para tarefas que precisam ser executadas apenas uma única vez.
- A classe `jade.core.behaviours.WakerBehaviour` implementa um comportamento que é disparado após um determinado intervalo de tempo. Em outras palavras, é uma classe abstrata com uma tarefa atômica, que é executada apenas depois que determinado intervalo de tempo expira.

A classe SimpleBehaviour implementa comportamentos atômicos, monolíticos e que nunca devem ser interrompidos. Dentre as classes derivadas pode-se citar a CyclicBehaviour, que representa um comportamento atômico que é executado sempre como se estivesse em loop infinito e TickerBehaviour, que implementa um comportamento que será executado periodicamente, e cuja frequência de repetições é definida pelo desenvolvedor.

A classe ReceiverBehaviour implementa um comportamento para recebimento de mensagens ACL sendo que o comportamento é encerrado quando uma mensagem ACL é recebida. Esta classe já implementa o método receive() como uma operação atômica, ou seja, o ato de receber uma mensagem é uma ação que nunca será interrompida durante sua execução.

A classe CompositiveBehaviour é uma classe abstrata para comportamentos compostos por outros comportamentos. Os comportamentos filhos são controlados por meio de alguma política de escalonamento, que divide a fatia de tempo de execução entre eles.

5.4.6 Comunicação entre Agentes na Plataforma JADE

A comunicação entre os agentes na plataforma JADE é realizada pela troca de mensagens com intuito de alcançar os objetivos de a sociedade de agentes. Portanto, a comunicação na plataforma JADE é essencial para que haja cooperação, colaboração ou negociação. Desta maneira, ocorre compartilhamento de conhecimento e a coordenação de ações entre os agentes.

Toda a troca de mensagens realizada no JADE é feita através de métodos próprios e com o uso de instâncias da classe ACLMessage, que possui um conjunto de atributos que estão em conformidade com as especificações da FIPA, implementando a linguagem FIPA-ACL. Em uma

mensagem do JADE o remetente é sempre um agente e o receptor pode ser um ou um conjunto de agentes.

De acordo com Gudwin 2004, FIPA-ACL é uma linguagem (ACL – Agent Communication Language) que viabiliza a comunicação entre agentes através de mensagens. Para tanto, a FIPA-ACL utilizam primitivas de comunicação, conhecida como atos comunicativos (a performativa como, por exemplo, INFORM, AGREE, QUERY, etc), para expressar crenças, incerteza, intenção, entre outros.

Para um agente enviar uma mensagem deve, primeiro, instanciar um objeto da classe ACLMessage, preencher os atributos com as informações necessárias e chamar o método send() da classe Agent. Por outro lado, para receber uma mensagem o agente deve fazer referência ao método receive() ou blockingReceive(). Os métodos para enviar e receber mensagens são implementados na classe Agent.

O envio e o recebimento de mensagens podem ser escalonados como atividades independentes do agente, através da adição dos comportamentos SenderBehaviour e ReceiverBehaviour na fila de comportamentos do agente.

O padrão FIPA define que somente as mensagens ACL são transportadas na plataforma, porém, não especifica nenhum padrão para o conteúdo da mensagem. Cada agente possui uma fila privativa de mensagens ACL a qual ele processa da forma que o mesmo achar mais interessante.

Quando um agente envia uma mensagem para o outro agente, essa mensagem vai para o DF (Diretório Facilitador) onde entra em uma fila e fica acessível a todos os agentes que consultam o DF.

JADE utiliza um mecanismo de transporte adaptável, ou seja, o mecanismo de transporte é selecionado de acordo com a situação da

plataforma. Isso ocorre com um objetivo único de atingir o menor custo possível de passagens de mensagens, as seguintes situações são possíveis:

- Se o agente receptor reside no mesmo container de agentes, então, o objeto java representante da mensagem ACL é passado para o receptor via eventos.
- Se o agente receptor reside na mesma plataforma JADE, mas em containeres diferentes, então, a mensagem ACL é enviada usando Java RMI.
- Se o agente receptor reside em uma plataforma de agentes diferente, então, o protocolo IIOOP (Internet Inter – ORB Protocol) é usado de acordo com o padrão FIPA.

A figura 14 mostra as possibilidades que JADE tem para realizar suas trocas de mensagens.

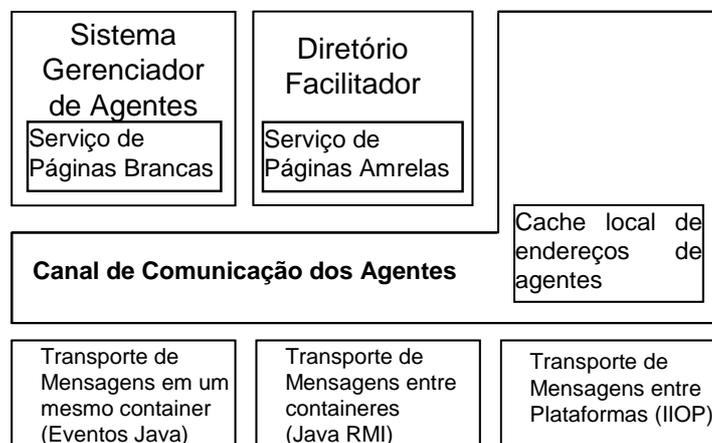


Figura 14: Formas de Comunicações entre Agentes

5.4.7 Ferramentas de Gerenciamento

A plataforma JADE consiste na disponibilidade de ferramentas gráficas que auxilia e facilita o gerenciamento/monitoração de ciclo de vida de aplicação e dão suporte a depuração. Através dessas ferramentas é possível gerenciar inclusive remotamente os agentes existentes na plataforma JADE.

As principais ferramentas gráficas da plataforma JADE são: RMA (Remote Monitoring Agent), Dummy Agent, Sniffer Agent, Introspector Agent, Log Manager Agent e DF (Directory Facilitator).

O gerenciador remoto de agentes (RMA) é uma interface gráfica para o controle e gerenciamento da plataforma JADE, que permite o controle dos estados do ciclo de vida de todos agentes em execução. Nesta interface gráfica estão centralizadas algumas funcionalidades como, por exemplo, chamar as outras ferramentas da plataforma JADE. A seguir na figura 15 mostra o gerenciador remoto de agente (RMA).

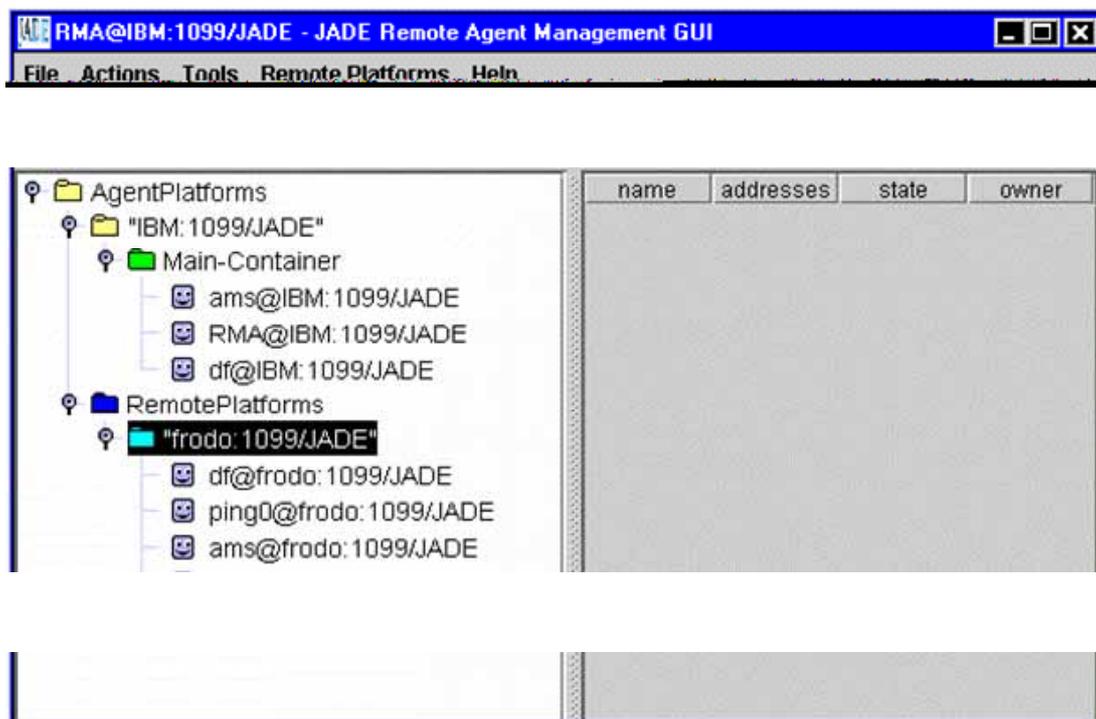


Figura 15: Interface Visual do RMA

5.5 Sistema Baseado em Conhecimento

Os sistemas de IA implementam, portanto, comportamentos inteligentes de especialistas humanos. Para tanto, utiliza-se uma abordagem cognitiva – Sistema Baseado em Conhecimento (SBC).

Segundo Newel (1982), SBC elabora sistemas diretamente através do conhecimento acerca dos processos, isto é, declarativamente, dessa forma os computadores podem realizar dedução automática através de inferências. Basicamente, o SBC fundamenta-se na separação entre o conhecimento e o processo dedutivo ou inferência. Conforme Newell a concepção de sistema dessa natureza passa por três especificações sucessivas, que são:

- I. O nível de conhecimento ou epistemológico, que consiste em uma especificação abstrata do conhecimento do domínio ou problema que se deseja modelar;
- II. O nível lógico, que converte as especificações de conhecimento em sentenças de lógica formal;
- III. O nível de implementação, que codifica estas sentenças em linguagem computacional, seja em uma linguagem ou em um banco de dados dedutivo, conhecido como base de conhecimento.

Portanto, observou-se, que um SBC é fundamental a existência do nível de conhecimento que guarda uma relação mais próxima e direta com o conhecimento sobre o domínio a ser modelado. O intuito do paradigma declarativo (o nível lógico) é modelar o raciocínio humano. Colocar o conhecimento do lado de fora da aplicação (entidade de software) é uma prática que viabiliza a portabilidade e flexibilidade do conhecimento e, então, independente de máquina.

5.5.1 Formalismo de Representação do Conhecimento

A representação do conhecimento é o estudo de como o conhecimento sobre o mundo pode ser representado e que tipo de raciocínio pode ser feito com aquele conhecimento. Portanto, a representação do conhecimento pretende levar o conhecimento de um domínio para uma entidade de software e dotá-la de alguma capacidade de raciocínio.

Em 1958 o pesquisador McCarthy no sistema Advice Taker começou a fomentar a idéia de manter fatos e conhecimento do lado de fora dos programas. Utilizando a linguagem Lisp na representação de fatos e regras em sistemas especialistas McCarthy percebeu-se que as regras poderiam ser mais genéricas se os fatos estivessem separados ao invés de colocar os fatos dentro das regras.

LISP (List Processing, em português Processamento de Listas), uma linguagem funcional voltada para o manejo de listas, foi criada justamente com o propósito de representar conhecimento de forma mais abstrata.

Em 1978, apareceram os primeiros sistemas de inferência orientados a padrões que incorporam a programação relacional ou lógica. Segundo Hayes-Roth et al. (1978), esses sistemas são compostos basicamente de três elementos, que são:

- I. Um formalismo fundamentado em lógica matemática, no qual o código do sistema baseado em conhecimento será escrito.
- II. Um método ou estratégia de resolução para o formalismo escolhido, conhecido como mecanismo de inferência.
- III. Estratégias de controle e escalonamento da inferência ou métodos de resolução de conflitos, implementados dentro dos mecanismos de

inferência por módulos chamados executivos, com a função de guiar e otimizar a inferência, que normalmente não é consistente ou completa.

5.5.1.1 Sistemas de Produção

Casamento: o sistema de produção faz o casamento dos fatos da memória de trabalho com as regras.

Resolução de Conflitos: escolhe um par <regra, fatos> do conjunto de conflitos de acordo com a estratégia de resolução de conflitos utilizada.

Disparo da Regra: executa as ações da regra escolhida no passo anterior com suas variáveis substituídas pelos fatos que tornaram a regra disparável.

Os sistemas de produção podem utilizar o algoritmo encadeamento progressivo e regressivo (Jackson, 1999). No encadeamento progressivo (forward-chaining), a inferência é feita de fatos que acredita-se que sejam verdadeiros para novos estados que as condições nos permitem estabelecer. No encadeamento regressivo (backward-chaining), o encadeamento se dá de um objetivo para as condições necessárias para que esse possa ser estabelecido. O motor de inferência que utiliza o encadeamento progressivo casa o lado esquerdo da regra com as variáveis da memória de trabalho, executando em seguida as ações definidas no lado direito. Por outro lado, o motor de inferência que utiliza o encadeamento regressivo casa o lado direito das regras com os objetivos os quais se tenta provar, gerando como novos sub-objetivos as condições definidas no lado esquerdo, até que se atinjam condições atômicas (fatos já presentes na memória de trabalho).

Outra vantagem dos sistemas de produção é que existem motores de inferência implementados em linguagens convencionais como, por exemplo, C e Java. Portanto, esses motores de inferência são integráveis a códigos imperativos e mantém o conhecimento fora dos programas. Desta forma, as regras de produção expressam uma separação ainda mais radical e eficiente entre dados.

5.6 JESS (Java Expert System Shell)

Em meados de 1997, o pesquisador Ernest J. Friedman-Hill do laboratório Sandia da Califórnia desenvolveu o sistema baseado em regras denominado de JESS (Java Expert System Shell).

JESS é um “script shell” para o desenvolvimento de sistemas especialistas baseados em regras e totalmente escrito na linguagem Java (FRIEDMAN-HILL, 2006). A versão mais recente do JESS é a 7.0.

A estrutura básica de uma aplicação baseado em regras desenvolvida em JESS é formada por uma base de regras e um motor de inferência. A base de regras representa o conhecimento do especialista humano de acordo com o domínio da aplicação que será construída. O motor de inferência realiza o processamento do conhecimento, no qual consiste em tirar conclusões a partir do conhecimento armazenado utilizando o encadeamento lógico das regras. O motor de inferência do JESS suporta tanto o encadeamento regressivo quanto o encadeamento progressivo.

A memória de trabalho – às vezes chamado de base de fatos – contém todos os pedaços de informação que JESS trabalha. A memória de trabalho pode assegurar as premissas e as conclusões de suas regras.

No encadeamento progressivo, conhecido também de encadeamento dirigido por dados, a parte do lado esquerdo das regras é casada com os fatos existentes na memória de trabalho. Quando ocorre o casamento das regras do lado esquerdo com os fatos a parte do lado direito da regra é executada, ou seja, uma ação é disparada que pode ocorrer a introdução de novos fatos na memória de trabalho.

Por outro lado, no encadeamento regressivo, conhecido também de encadeamento dirigido por objetivos, uma lista de objetivos controla o

comportamento do sistema. Um determinado objetivo pode ser satisfeito diretamente por um elemento da memória de trabalho, ou podem existir regras que possuam a descrição desse objetivo em suas partes direitas, permitindo a inferência do objetivo em questão. Se uma regra satisfaz essa condição, sua parte esquerda é adicionada à lista de objetivos correntes. Caso uma regra tenha todas as suas condições satisfeitas diretamente pela memória de trabalho, o objetivo em sua parte direita é adicionado à memória de trabalho. Um objetivo é abandonado caso não possa ser satisfeito diretamente pela memória de trabalho, nem inferido por alguma regra.

O motor de inferência do JESS utiliza o algoritmo Rete com a finalidade de uma maior eficiência no processamento de regras. O funcionamento desse algoritmo se dá pela construção de uma rede de nós, onde cada nó representa testes feitos em cima do lado esquerdo das regras. Os fatos adicionados ou removidos da memória de trabalho são/foram processados pela rede até atingirem a base, onde estão as regras individuais. Se um conjunto de fatos processado pela rede passa por todos os testes sobre o lado esquerdo de uma regra em particular, o lado direito dessa regra é executado, caso a remoção de um dos fatos da memória de trabalho não invalide a regra.

Entre as facilidades providas, o JESS permite a integração de sua base de conhecimento, sua linguagem de scripts e sua máquina de inferência ao ambiente Java e suas tecnologias.

O JESS, também, pode utilizar a tecnologia javaBeans para permitir que as regras tenham acesso aos métodos `get()` e `set()` das instância de uma classe que segue o padrão bean. Logo o motor de inferência do JESS pode ler e alterar os atributos dessa classe.

JESS armazena o conteúdo de memória de trabalho usando um conjunto de índices personalizado que observa um pedaço de informação

particular muito rápido. Cada item na memória de trabalho aparece sobre esta lista na ordem em que foi adicionado.

Portanto, uma das características importante do JESS é que este pode ser integrado em um agente desenvolvido na plataforma JADE capacitando esse agente com a capacidade de inferências. A plataforma JADE, por sua vez, facilita essa integração oferecendo um comportamento chamado JessBehaviour, onde o JADE provê um shell para o agente, garantido sempre que possível a conformidade com as regras da FIPA, enquanto o JESS realiza todo o raciocínio necessário para o agente.

6 PROPOSTA DE TRABALHO

Esta seção descreve o funcionamento do modelo de o mecanismo desenvolvido para a efetivação de apoio ao professor no processo de acompanhamento do aprendiz no ambiente de ensino-aprendizagem colaborativa à distância, o NetClass. Este mecanismo faz uma análise das informações do aprendiz, derivando e concluindo novas informações e elabora uma mensagem ao aprendiz contendo a orientação. O professor recebe esta orientação podendo aceitar ou não ou até mesmo fazer observações.

6.1 Mecanismo de Apoio no Acompanhamento do Aprendiz (MAAA)

Em suma, o acompanhamento do aprendiz, como foi visto no capítulo anterior, envolve a tarefa de o professor acompanhar o aprendiz pela observação de suas ações no seu ambiente e pelas informações de suas atividades com a finalidade de avaliar o desempenho, grau de interesse e outros aspectos ligados ao aprendiz, de acordo com os critérios de avaliação adotada pela coordenação pedagógica ou professor de um curso.

Nos ambientes que promovem a Educação a Distância o acompanhamento do aprendiz é alcançado pelos mecanismos que exploram o Log, este armazenam as informações das atividades e as ações do aprendiz em forma de registro em sistema de arquivos ou banco de dados. Essas ações são realizadas através das trocas de mensagens textuais viabilizadas pelas ferramentas de comunicação, tais como: fóruns, chats, mural, dentre outros.

Várias tecnologias têm sido utilizadas para proverem meios mais elaborados de extrair informações de interesse do professor pelos mecanismos a partir dos registros conforme as métricas definidas no algoritmo do mecanismo. Os primeiros mecanismos utilizados para explorar esses registros (os logs) no processo de acompanhamento do aprendiz apresentavam os

resultados de um modo bem rudimentar. Com o uso das tecnologias de agentes artificiais nesse processo de acompanhamento obteve-se um aprimoramento na apresentação e organização dos resultados. Uma significativa melhora foi a inserção das técnicas de mineração de dados que classificam, associam ou agrupam as informações do aprendiz fornecendo, então, resultados mais elaborados.

Contudo, mesmo com o avanço obtido com a utilização dessas técnicas advinda da Inteligência Artificial, aplicação das técnicas de agente artificial e mineração de dados, o processo de acompanhamento do aprendiz, ainda, carece de aperfeiçoamento devido à sobrecarga de tarefas destinadas para o professor e, também, pela natureza de critérios de avaliação. Esse processo de acompanhamento, que envolve a observação das informações geradas pelas ações do aprendiz na resolução de atividades pelos meios de comunicação existentes, demanda, ainda, muito esforço do professor, pois este tem que examinar manualmente cada resultado proveniente de mecanismos para que possa tomar uma decisão, isto é, elaborar uma orientação. Já a natureza de critérios de avaliação é um fator inerente a cada professor, pois cada professor tem uma maneira peculiar de avaliar o aprendiz. Nos mecanismos de exploração de registros pesquisados há a necessidade de pré-definir, durante a fase de desenvolvimento do mecanismo, um conjunto de métricas que representem como os registros serão explorados, ou seja, as informações que serão extraídas e como apresentar essas informações para o professor.

O apoio à tomada de decisão ao professor no acompanhamento do aprendiz, neste trabalho, ocorre quando o mecanismo proposto relaciona as informações dispersas pertencentes ao aprendiz no ambiente do professor e, de acordo com a análise, dispara uma ação coerente (a orientação) com o propósito de auxiliar o professor no acompanhamento do aprendiz. Desta forma, esse mecanismo realiza a tarefa de o próprio professor solicitar informações do aprendiz (os resultados) em seu ambiente para relacionar as

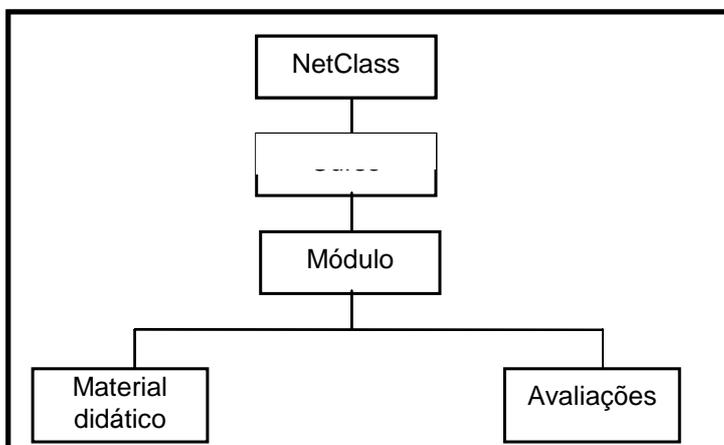
mesmas com o objetivo de orientar o aprendiz. Esse é o objetivo do Mecanismo de Apoio no Acompanhamento do Aprendiz (MAAA), composto de uma sociedade de agentes inteligentes, constituída de agentes reativos e cognitivos, baseado em técnicas provenientes da Inteligência Artificial.

O MAAA é aplicado ao NetClass, um ambiente de ensino-aprendizagem à distância, baseado no princípio de Inteligência Artificial Distribuída e o paradigma da Aprendizagem Colaborativa. O NetClass (Labidi, 2003), é um ambiente de aprendizagem colaborativa assistida por computador e faz uso das tecnologias baseadas na web. Nesse ambiente atuam as seguintes entidades: o aprendiz, o professor (responsável por um ou mais módulos), o monitor (auxiliar do professor) e o coordenador (administra os cursos). Cada uma dessas entidades possui um ambiente individualizado com todas as funcionalidades pertinentes a sua categoria. A comunicação entre ambientes, ou seja, a interação entre professor/aprendiz, aprendiz/professor e outras entidades, se opera através das ferramentas de comunicação disponíveis – tais como o chat, o mural e o fórum – que ajudam o aprendiz na resolução de atividades. O ambiente NetClass pode ser visto na figura 16.



Figura 16: Ambiente NetClass

O sistema de ensino no NetClass é composto de cursos; cada curso é formado de módulos (correspondente às disciplinas); cada módulo é composto de material didático (lições) e avaliações (provas e testes), como mostra a figura 17 abaixo. O material didático fornece ao aprendiz subsídios para sua construção de conhecimento. A base pedagógica do NetClass é centrada no aprendiz e na aprendizagem em grupo, onde cada aprendiz deve compartilhar idéias, dúvidas e opiniões com seus companheiros, buscando aprimorar seus conhecimentos e aplicá-los na resolução de atividades, processo esse que faz surgir o conhecimento.



Todas as atividades do aprendiz no ambiente NetClass são registradas, gerando um grande volume de informações armazenadas em banco de dados, caracterizando o que se costuma chamar de log. O NetClass, como a maioria dos ambientes colaborativos (por exemplo: Moodle, TelEduc, Eureka, Academusnet, AulaNet, entre outros), tem mecanismos que exploram os logs (as informações dos aprendizes) e apresentam o resultado para o professor em formato de tabelas.

6.1.1 Acompanhamento do Aprendiz no Ambiente NetClass

Tendo em vista todas as questões que foram mencionadas ao longo desse trabalho, relacionadas aos mecanismos existentes no acompanhamento do aprendiz, realizou-se um estudo de caso aplicado ao NetClass. Nesse estudo de caso formou-se o Curso de Ciência da Computação constituído por dois módulos – Programação Orientada a Objetos e Estrutura de Dados – com 20 aprendizes em cada um. Cada módulo tem seu respectivo professor como mostra a figura 18.

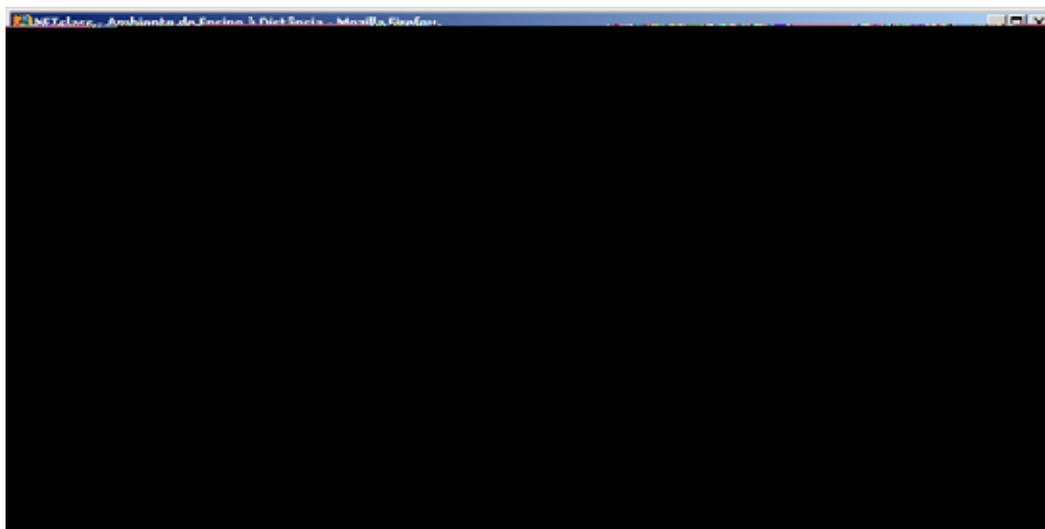


Figura 18: Módulos de um curso no ambiente NetClass

Observou-se, diante do exposto na seção anterior, que os mecanismos, atualmente existentes na literatura, para a consolidação de uma avaliação informal, se resumem ao esquema expresso na figura 19.

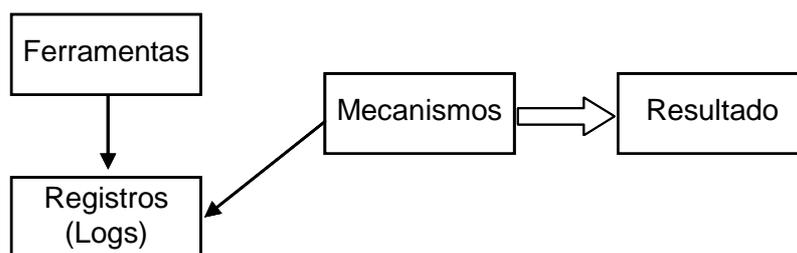


Figura 19: Esquema dos Mecanismos de Avaliação Informal existentes

Seguindo esse esquema, as ferramentas (meios de comunicação) utilizam uma forma de armazenamento que constituem os registros. Os mecanismos de avaliação informal exploram esses registros de acordo com determinadas métricas, que foram definidos estaticamente em tempo de desenvolvimento do mecanismo, e produzem os resultados para o professor, em forma de tabelas, gráficos ou, até mesmo, sem nenhum tratamento organizacional. O professor, para descobrir o motivo de um desempenho ruim nas atividades, tem a necessidade de co-relacionar as informações dos resultados providas pelo sistema para ser capaz de realizar uma orientação para o aprendiz a fim de melhorar o processo de ensino-aprendizagem.

Portanto, o acompanhamento do aprendiz em suas atividades no ambiente NetClass é expressa conforme a figura 20 que representa o funcionamento deste esquema.

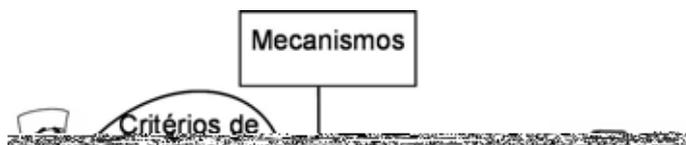


Figura 20: Acompanhamento do Aprendiz no NetClass

No NetClass tem mecanismo para acompanhar o desenvolvimento/desempenho do aprendiz nas atividades; o mecanismo extrai informações dos logs de acordo com as métricas estabelecidas no tempo de desenvolvimento do próprio mecanismo e gera um resultado para o professor; o professor, por sua vez, para fazer uma orientação ao aprendiz precisa relacionar as informações fornecidas pelos mecanismos; essa co-relação de informações do aprendiz fornece ao professor subsídios para uma orientação precisa e o professor a realiza conforme os critérios de avaliação.

Um dos resultados do acompanhamento do estudo de caso com o NetClass, é que, realmente, o professor consome muito tempo explorando e analisando o seu ambiente em busca de subsídios capazes para fazer uma orientação consistente aos seus aprendizes. Observou-se, também, que o processo de tomada de decisão é de caráter exclusivo do professor, feito manualmente e sem suporte de qualquer mecanismo, pois o conhecimento para essa tomada de decisão está armazenado no próprio professor, em seu cérebro, proveniente de sua experiência e das orientações transmitidas (no caso os critérios de avaliação) pela coordenação do curso.

Desta forma, para que se possa proporcionar um suporte flexível para o professor no processo de acompanhamento do aprendiz, o presente trabalho propõe a introdução de um mecanismo de inferência no tradicional esquema, mostrado na figura 19. Para tanto, será utilizado um agente cognitivo que contém, além do motor de inferência (JESS, 2006), uma base de conhecimento (JESS, 2006) que reterá o conhecimento necessário para a elaboração da orientação. A figura 21 apresenta o novo esquema, adaptando-se o esquema presente na figura 19 acrescido de novo elementos.

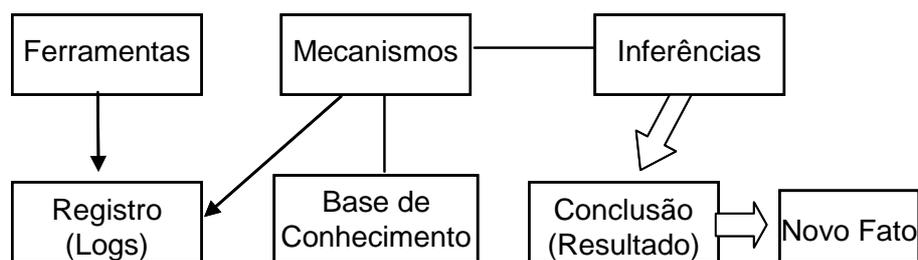


Figura 21: Novo Esquema proposto de Avaliação Informal do MAAA

Nesse novo esquema, projetado no MAAA, os critérios de avaliação passam a ser inseridos em uma base de conhecimento. Com a capacidade de inferência o MAAA relaciona os fatos (informações do aprendiz) e casa-os com os critérios (como o professor avalia o aluno), o que ocasiona alguma decisão/orientação e/ou dedução de um novo fato.

Em geral, nos ambientes virtuais, cada professor tem sua maneira de avaliar e acompanhar um aprendiz de um curso entende-se que podem ser adotadas estratégias de acompanhamento e avaliação (ALMEIDA, 2003; APARECIDA, 2002). No contexto do MAAA, as estratégias dos professores servem para identificarem problemas no processo de acompanhamento do aprendiz para, então, ajudá-lo com uma orientação condizente.

O MAAA fornece um valioso auxílio para o professor na orientação ao aprendiz, mostrando, portanto, os possíveis problemas que o aprendiz pode está passando. Para realizar o modelo de MAAA analisou-se como é feito todo processo de acompanhamento do aprendiz no ambiente NetClass de ensino-aprendizagem, sintetizado no esquema da figura 22.

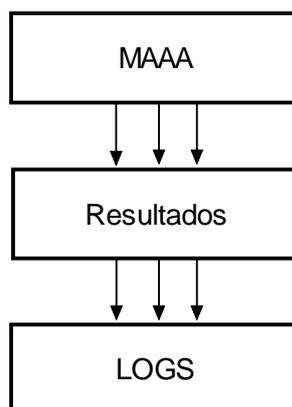


Figura 22: Esquema do MAAA

Antes, havia basicamente duas camadas. A primeira camada representava os logs, na qual ficam armazenados as ações realizadas pelo o aprendiz e os conceitos de atividades, provas e participações em chats e fóruns e demais participações.

Na segunda camada, os resultados são gerados pelos mecanismos existentes esquematizados conforme a figura 19 que auxiliam o professor no acompanhamento do aprendiz somente coletando e organizando as informações e apresentando estas em forma de tabelas em HTML. Todavia, os resultados fornecem uma análise quantitativa das informações do aprendiz. O professor, então, analisa esses resultados manualmente para buscar meios de identificar as possíveis causas de algum resultado abaixo do esperado e, assim, agir adequadamente para reverter o quadro.

Agora, com a introdução da terceira camada o MAAA abstrai a tarefa de o professor analisar os resultados, manualmente, possibilitando, portanto, um suporte flexível na orientação do aprendiz.

6.1.2 Os Objetivos do MAAA

Com base na explanação e na justificativa mostrada de como sucede o acompanhamento das atividades do aprendiz no ambiente NetClass foi definido, em primeiro lugar, o objetivo geral do modelo do mecanismo MAAA.

O objetivo global do mecanismo do MAAA é observar continuamente os resultados oriundos dos logs e, dependendo das informações de algum aprendiz em alguma atividade, utilizar seu próprio conhecimento armazenado na sua base de conhecimento, que são os critérios de avaliação que expressam como o professor deve acompanhar e avaliar o aprendiz, citadas anteriormente. Dessa forma, por exemplo, se a nota do aprendiz for insuficiente (conceito ruim), regular ou bom o MAAA pode correlacionar as informações dispersas do aprendiz com sua capacidade de inferências para, então, auxiliar o professor na orientação ao aprendiz. Após o mecanismo já ter relacionado as informações, o professor vai receber uma mensagem indicando as possíveis causas do desempenho do aprendiz. O professor antes de liberar a mensagem ao aprendiz pode também fazer as suas considerações e observações.

Para a concretização do modelo do MAAA, definiu-se os seus objetivos específicos, que são os seguintes:

- 1) Explorar continuamente (ou a cada sessão) os logs (base de dados) para “saber” o desempenho do aprendiz nas provas, o conceito nas contribuições em chats e fóruns e o acesso as lições on-line e off-line, convertendo os dados/informações do aprendiz em fatos e inserindo-os na base de conhecimento;
- 2) Capturar o conhecimento do professor (como o professor efetua o acompanhamento do aprendiz), através de uma interface amigável, para

3) Prover um meio de capacidade de raciocínio sobre a base de conhecimento;

4) Levantar situações problemáticas, as possíveis causas e elaborar uma mensagem contendo a orientação para o aprendiz e enviá-la ao professor, que poderá referendar ou não a orientação;

O MAAA faz o acompanhamento do aprendiz tanto nas lições, que são as atividades elaboradas pelo professor no decorrer do curso, como nas aplicações de provas e testes com a finalidade de auxiliar o professor. Para tanto, o MAAA usufrui do conhecimento do professor, que são os critérios de avaliação definidos pela coordenação do curso. Portanto, faz-se necessário definir os critérios de avaliação que simboliza como o professor toma uma decisão. Entretanto, conforme mostram as pesquisas (KEMCZINSKI, 2005; RESENDE, 2002) não há um conjunto padrão de como o professor avalia e/ou acompanha o aprendiz, pois cada instituição define seu conjunto de critérios de avaliação. Devido a esse fato, o MAAA utilizará um conjunto de critérios de avaliação definidos pela coordenação do ambiente NetClass, já que a definição dos mesmos abre o leque para outros estudos. O intuito é mostrar que utilizando esse mecanismo com a capacidade de inferências baseado nesses critérios de avaliação pode-se automatizar a tarefa de acompanhamento do aprendiz a qual resultará em uma tomada de decisão e, assim, diminuir a sobrecarga de atividades no professor.

6.1.3 Os Critérios de Avaliação

No ambiente NetClass é atribuído um conceito das provas ao aprendiz conforme a tabela 5. Essa atribuição de conceito é de autoria do professor. De acordo com intervalo da nota, o aprendiz pode receber os seguintes conceitos: insuficiente, ruim, bom ou excelente.

Tabela 5: Métricas da Prova

Nota	Conceito
0 – 4,9	Ruim
5,0 – 6,9	Regular
7,0 – 8,9	Bom
9,0 – 10	Excelente

As participações em fóruns e chats do aprendiz são acompanhadas pelo monitor de o ambiente NetClass que, conforme a relevância da contribuição, atribui uma nota. Uma sessão de chat e participação/contribuição em fórum é referente a uma avaliação e quem define a data e à hora de uma sessão de chat é o professor do curso. No final de cada sessão de chat o aprendiz recebe uma nota pela sua participação. O conceito do aprendiz é a média da soma das notas dividida pelo o número de sessões de chat ou fórum de acordo com a tabela 6:

Tabela 6: Métricas de contribuição em Fóruns e Chats

Contribuição	Conceito
0 – 4,9	Ruim
5,0 – 6,9	Regular
7,0 – 8,9	Bom
9,0 – 10	Excelente

Também, as interações do aprendiz com seu ambiente podem mostrar o grau de interesse e de participação do aprendiz através da frequência de acesso ao módulo, se o aprendiz fez o download da lição off-line do seu módulo e o número de vezes que o aprendiz acessou a lição on-line do seu módulo. Nas tabelas 7, 8 e 9, são mostradas, respectivamente, as métricas da interações do aprendiz com seu ambiente.

Tabela 7: Métricas das interações do aprendiz com o módulo

Frequência de acesso ao módulo	Conceito
<= 25%	Ruim
> 25% e < 50%	Regular
>=50% e <= 75%	Bom
> 75%	Excelente

Tabela 8: Métricas de download das lições off-line

Download das lições off-line	Conceito
<= 25%	Ruim
> 25% e < 50%	Regular
>=50% e <= 75%	Bom
> 75%	Excelente

Tabela 9: Métricas de acesso às lições on-line

Acesso as lições on-line	Conceito
<= 25%	Ruim
> 25% e < 50%	Regular
>=50% e <= 75%	Bom
> 75%	Excelente

Para dotar um mecanismo com a capacidade de inferências, de derivar e de relacionar as informações do aprendiz, é preciso, previamente, definir o conhecimento que constituirá o mecanismo, ou seja, a base de conhecimento. A intenção, portanto, é transferir esses critérios de avaliação para uma base de conhecimento. Essa base de conhecimento representa o conhecimento propriamente dito de como o professor orienta o aprendiz e, também, permite um meio para realizar o raciocínio derivando, portanto, novas informações do aprendiz. É importante destacar, que, antes, o conhecimento do professor (critérios de avaliação) não era representado em software. Agora, esse conhecimento será representado explicitamente em uma base de conhecimento, que proporcionará uma forma de raciocínio.

6.1.4 Arquitetura do MAAA

Este subcapítulo descreve-se as tecnologias e arquitetura dos Agentes Inteligentes que atuam no processo de acompanhamento das atividades dos aprendizes no ambiente NetClass de ensino-aprendizagem. É utilizada a plataforma JADE (JADE, 2006) e o JESS (Java Expert System Shell) (JESS, 2006). Retratam-se os Agentes Inteligentes que constitui o MAAA, enfatizando seus comportamentos e como eles fazem para alcançar seus objetivos, sendo um agente cognitivo, o Agente de Suporte ao Professor, e três agentes reativos, que são: Agente Monitor, Agente Conversor e Agente de Aquisição de Conhecimento.

A arquitetura do MAAA, conforme a figura 23 mostra que este mecanismo é constituído de uma Sociedade de Agentes Inteligentes, uma Base de Conhecimento e uma Base de Dados.

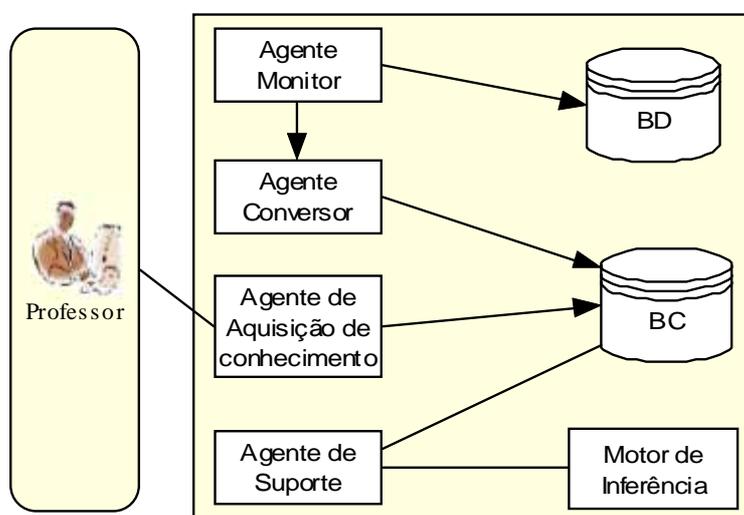
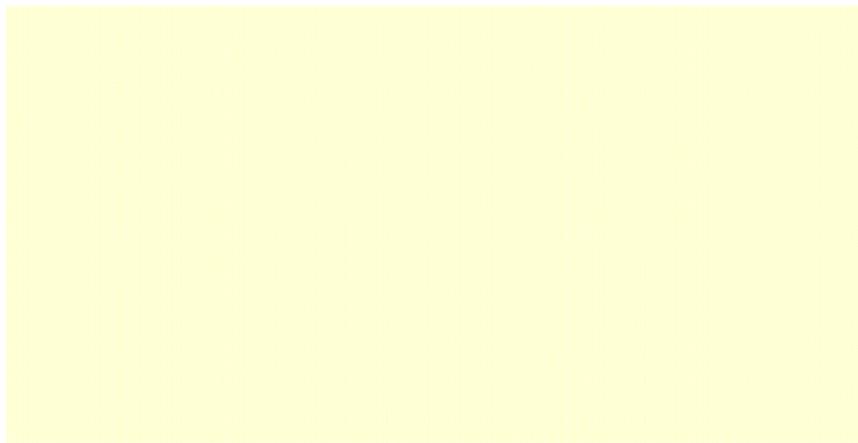


Figura 23: Arquitetura do Modelo do MAAA

O Banco de Dados é responsável pelo armazenamento dos registros (logs) das atividades do aprendiz e informações gerados pelas ferramentas de comunicação. A Base de Conhecimento armazena as regras (o conhecimento do especialista, no caso o professor) e os fatos (as informações referentes ao aprendiz) com o objetivo de dotar o MAAA de capacidade de raciocínio para a

tomada de decisão, ou seja, a orientação. A base de conhecimento do Mecanismo de Apoio no Acompanhamento do aprendiz irá funcionar segundo o esquema apresentado na figura 24.



inteligentes, sendo três reativos e um cognitivo, que desempenham os seguintes papéis:

- Monitor
- Conversor
- Aquisição de Conhecimento
- Suporte ao Professor

Nesta situação, a autonomia dos agentes possibilita a realização de sua tarefa em vista de alcançar o objetivo do mecanismo (acompanhar os aprendizes) através de execução de comportamentos. Portanto, com essa característica os agentes inteligentes desempenham suas atividades e gerenciam os grandes volumes de dados das atividades gerados pelo aprendiz.

Utiliza-se o Framework JADE para o desenvolvimento dos agentes inteligentes supracitados porque a plataforma fornece bibliotecas prontas, o ambiente de execução e ferramentas gráficas para gerenciamento da plataforma e conformidade com o padrão FIPA (FIPA 2006), facilitando, assim, o desenvolvimento desses agentes.

- O ambiente de tempo de execução é o local onde os agentes instanciados moram e que necessita ser ativo sobre um determinado host antes de um ou mais agentes ser executados.
- A biblioteca de classes que programadores podem usar (diretamente ou por especialização) para desenvolver os agentes.
- As ferramentas gráficas da plataforma permitem a administração e o monitoramento das atividades de agentes executados.

Portanto, JADE é um middle-ware que facilita o desenvolvimento de Sistema Multiagentes implementando os serviços básicos e infra-estrutura de aplicação distribuída Multiagentes, tais como: Ciclo de vida e mobilidade do agente, serviços de páginas amarelas e brancas, passagem e transporte de mensagem ponto a ponto, segurança do agente, programação de tarefas de agentes e um conjunto de ferramentas gráficas para monitoração e depuração. Então, a aplicação baseada em JADE é composta de vários componentes ativos chamados de agentes.

Os agentes inteligentes que compõem o MAAA trabalham de maneira integrada para alcançar seu objetivo. O diagrama de classe dos agentes pode ser visto na figura 25.

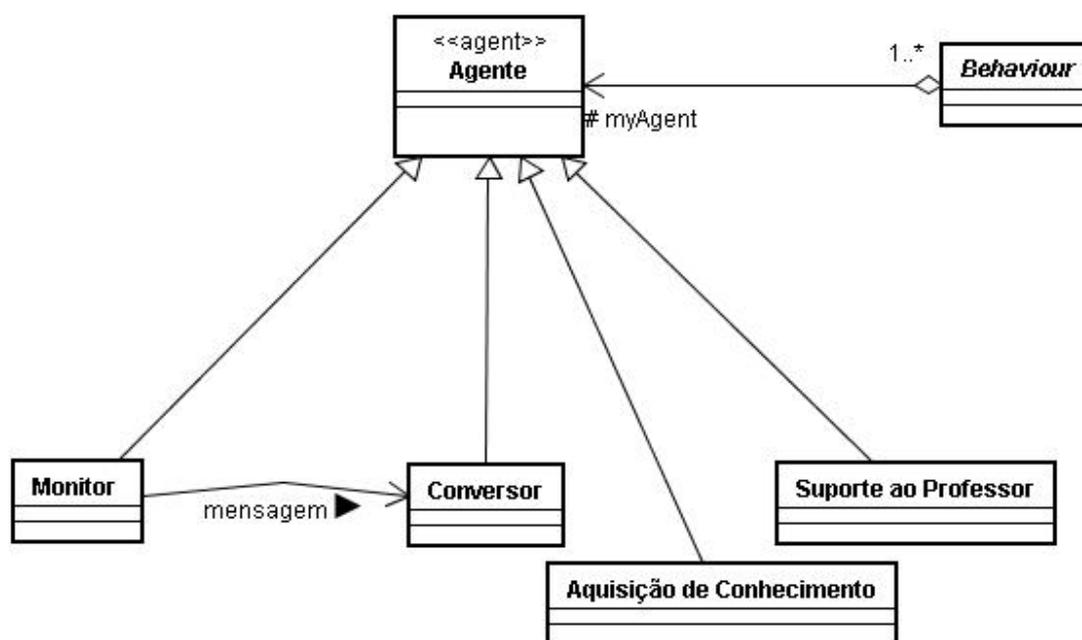


Figura 25: Diagrama de Classe dos Agentes Inteligentes do MAAA

Cada agente Inteligente do MAAA no ambiente JADE realiza seu trabalho através de comportamentos (*Behaviour*) que, geralmente, representa a tarefa a ser realizada.

6.1.5.1 O Agente Monitor

Agente Monitor: esse agente reativo explora continuamente as atividades do aprendiz, que são armazenados em uma base de dados, para “saber” o desempenho do aprendiz nas provas, o conceito nas contribuições em chats e fóruns e o acesso às lições on-line e off-line. A figura 26 mostra o diagrama de caso de uso deste agente.

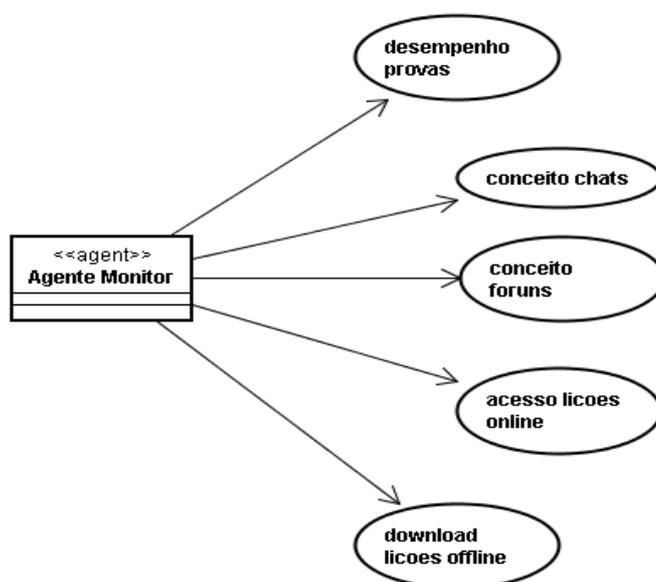


Figura 26: Caso de uso do Agente Monitor

O agente monitor periodicamente acessa a base de dados para extrair as informações do aprendiz e, posteriormente, caso haja alguma mudança – informação nova – é enviada uma mensagem para o agente conversor. Assim que o agente conversor recebe a mensagem do agente monitor contendo a informação do aprendiz o mesmo converte-as em fatos JESS e armazena na base de conhecimento.

6.1.5.2 O Agente Conversor

Agente Conversor: esse agente reativo converte as informações do aprendiz, extraídas na fase anterior, em fatos para inseri-los na base de conhecimento. A figura 27 mostra o diagrama de caso de uso deste agente.

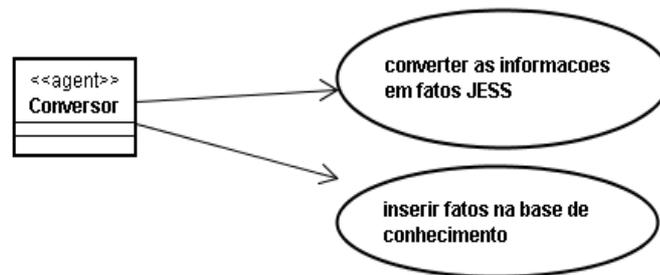


Figura 27: Caso de uso do Agente Conversor

O agente monitor fica inspecionando de tempo em tempo a base de dados com o objetivo de extrair informações do aprendiz e se comunica com o agente conversor através de troca de mensagem que aciona um comportamento específico para esta mensagem. A ação de o comportamento que o agente realiza é o método que é executado quando um agente recebe a mensagem. A figura 28 mostra o diagrama de seqüência do agente conversor.

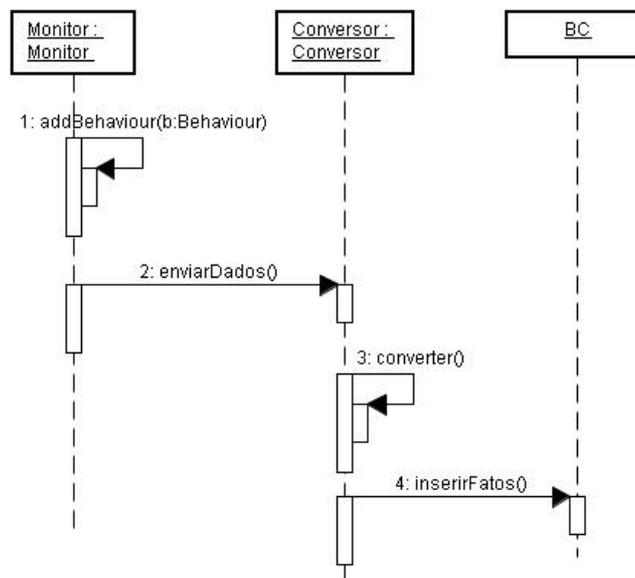


Figura 28: Diagrama de Seqüência

JADE fornece o mecanismo de troca de mensagens bastante flexível e eficiente, no qual o ambiente JADE cria e gerencia uma fila de mensagens ACL (Agent Communicaton Language) (FIPA, 2006) privada para cada agente. Dessa forma, cada agente tem autonomia sobre a sua fila de mensagens tratando-as da forma mais conveniente.

6.1.5.3 O Agente de Aquisição de Conhecimento

Agente de aquisição de conhecimento: Os critérios de avaliação, como o professor observa o aprendiz, foram transmitidos (capturados) para a base de conhecimento com o auxílio desse agente reativo. Através de uma interface amigável adequada, o professor fornece para o sistema seus critérios de avaliação. Esse agente inteligente recolhe essas informações e insere-as na base de conhecimento JESS. Esse item, que é a aquisição de conhecimento (os critérios de avaliação) do professor, fornece flexibilidade e dinamicidade no suporte de acompanhamento do aprendiz. A figura 29 mostra o diagrama de caso de uso deste agente.

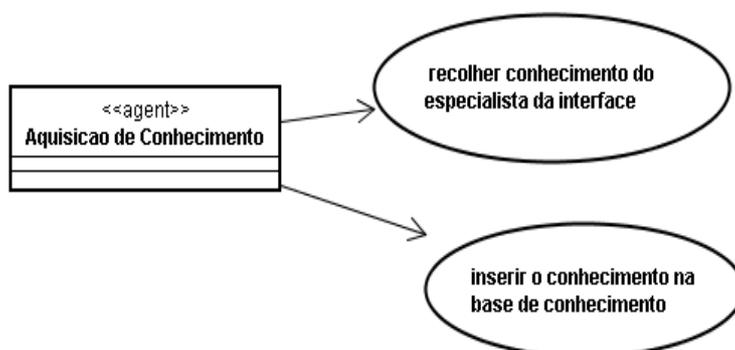


Figura 29: Caso de uso do Agente Aquisição de Conhecimento

Este agente reativo fica permanentemente escutando a interface gráfica, no qual o professor especialista usa para colocar seu conhecimento, com a finalidade de obter as informações provenientes da interface e convertê-las em regras JESS para armazenar na base de conhecimento.

6.1.5.4 O Agente de Suporte ao Professor

Agente de Suporte ao Professor: o sistema baseado em regras JESS possui um motor de inferência que aplica as regras aos fatos. Esses fatos constituem a memória de trabalho que representa o estado de uma situação e as regras simbolizam o conhecimento heurístico de um especialista humano. O casamento das regras sobre a memória pode disparar uma ação, a tomada de decisão, que vai ser realizado por este agente cognitivo. Ou seja, a tomada de decisão constitui a ação realizada (a orientação) de acordo com o casamento das regras (os critérios de avaliação) na base de conhecimento com a memória de trabalho onde ficam armazenados os fatos. Geralmente, o acompanhamento do aprendiz, ou seja, a observação das ações e das atividades realizadas do aprendiz ocasiona uma orientação. A figura 30 mostra o diagrama de caso de uso deste agente.

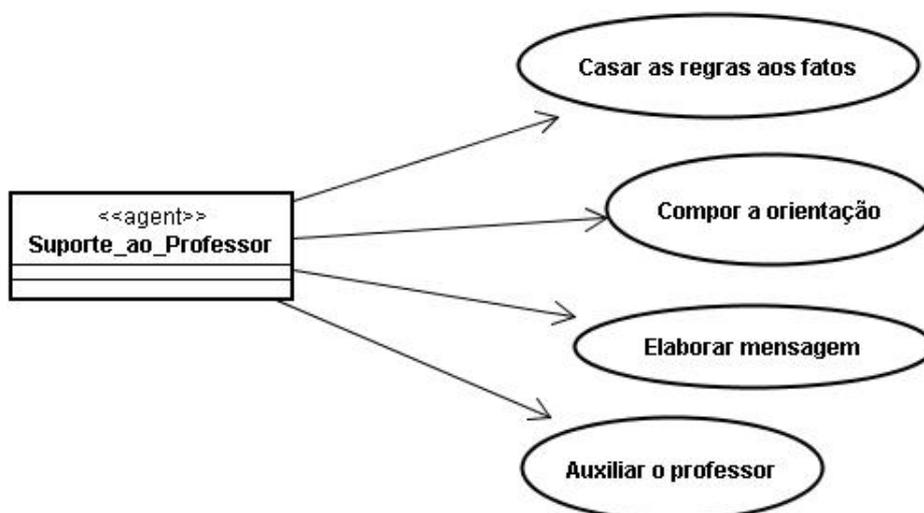


Figura 30: Caso de uso do Agente de Suporte do Professor

O agente de suporte ao professor é o único entre os demais agentes com a capacidade de inferência tornando-o um agente cognitivo. Este agente possui a capacidade de inferência, pois utiliza uma base de conhecimento e o motor de inferência aplicando, portanto, as regras aos fatos. O casamento das regras sobre os fatos ocasiona uma tomada de decisão que constitui a ação realizada (a orientação).

Para o desenvolvimento de agentes cognitivos no ambiente JADE utilizou-se o JESS (Java Expert System Shell). O JESS (JESS, 2006) é um motor de inferência implementado em JAVA para comandos em linguagem lógico-declarativa, que pode ser usado tanto como linha de comando, onde o usuário faz a entrada de fatos e regras de inferência.

Este agente cognitivo embutiu o motor JESS codificado em forma de um comportamento que se executa a cada inserção de novos fatos.

É importante salientar que as regras representam o conhecimento do professor especialista e a coleção de fatos representa as informações das atividades do aprendiz que é armazenado como lista, como mostra a figura 31.

FATOS JESS

(frequenciaModulo aprendiz1 ruim)
(nota aprendiz1 insuficiente)
(contribuicaoChat aprendiz1 bom)
(contribuicaoForum aprendiz1 bom)
(downloadLicaoOffline aprendiz1 ruim)
(acessoLicaoOnline aprendiz1 bom)
(frequenciaModulo aprendiz1 ruim)

Figura 31: Fatos JESS

O motor de inferência que foi embutido nesse agente é quem faz todo o processo de aplicar as regras aos fatos. As regras JESS têm duas partes: lado esquerdo e lado direito. O lado esquerdo consiste do padrão e o lado direito consiste de chamada de função. As duas partes da regra são separadas pelo símbolo => que pode ser lido como então. A regra em JESS é representada na figura 32 abaixo.

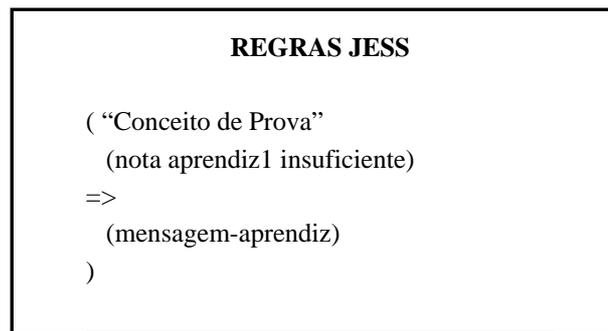


Figura 32: Regras JESS

O lado esquerdo da regra (equivalente a instrução if) assegura o padrão que é casado com os fatos na base de conhecimento. O lado direito da regra é feito de chamada de função.

Portanto, a aplicação de regras ao conjunto de fatos produz o processo de inferência e dedução simulando, então, a inteligência do MAAA.

6.1.6 O processo de tomada de decisão do MAAA

Os critérios de avaliação, o conhecimento do professor, que são representados pelo um formalismo na base de conhecimento, foram esquematizados de acordo com o diagrama da figura 33 abaixo. Além de exemplificar como o conhecimento do professor está organizado na base de conhecimento juntamente com os fatos agiliza a busca na base de conhecimento.

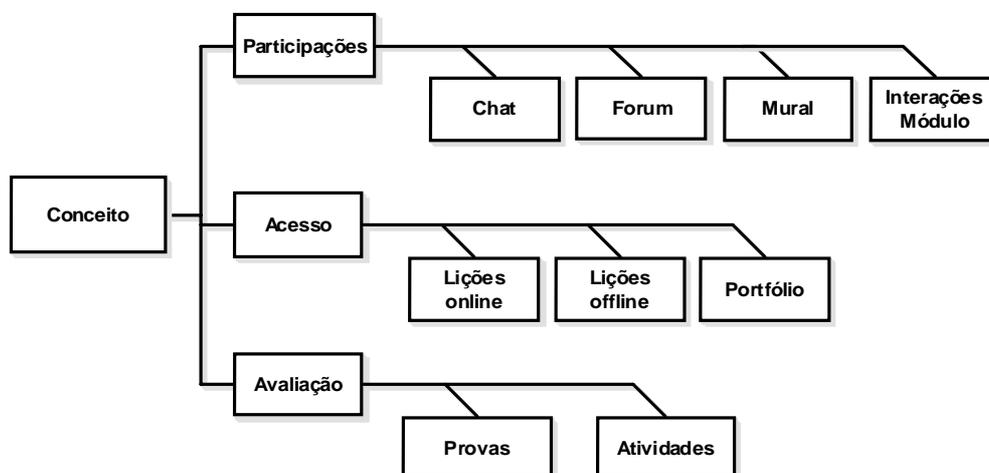


Figura 33: Organização da base de conhecimento em módulos

Assim, o MAAA faz à análise dos conceitos na sua base de conhecimento gerando, então, as ações (orientação) necessárias caracterizando a tomada de decisão. Em uma análise podem existir diferentes situações em função do conceito do aprendiz. Para cada situação é feita uma combinação com os demais conceitos como a participação, o acesso e a avaliação, co-relacionando as informações para buscar, portanto, subsídios para uma tomada de decisão. A figura 34 mostra algumas situações resultantes de uma análise e as possíveis ações para tratá-las. Os retângulos representam as informações do aprendiz coletadas pelo agente monitor e adicionado como fato pelo agente conversor e as elipses indicam o resultado da análise que ocorre pela aplicação das regras aos fatos na base de conhecimento.

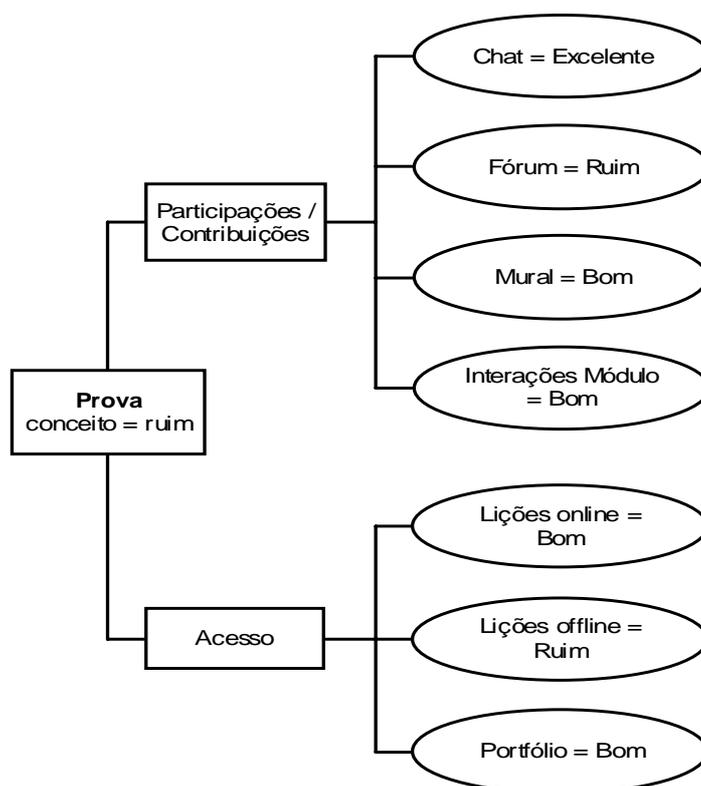


Figura 34: Esboço do Resultado com MAAA

Com base da análise na figura 34 acima, o MAAA, co-relaciona os conceitos e deriva novas informações e confecciona uma mensagem contendo a orientação. O professor recebe, portanto, essa mensagem com a orientação podendo alterar os campos. A mensagem é mostrada na figura 35.

ORIENTAÇÃO	
Prof. Nome	Nome do Aluno
Situação: • Conceito na prova de Estrutura de dados é ruim	
As possíveis causas: • Fóruns foi abaixo do esperado. • Chats foi bom • Interações com o Módulo foi bom • Lições online foi bom • Lições offline foi ruim • Portfólio foi Bom	
Orientação: Você precisa participar mais do Fórum, melhorar sua participação no Chat; Quase não baixou as lições off-line importantíssima para o entendimento sobre pilhas. Com problemas para acessar o material didático? O conceito ruim reflete no de acesso ao material didático e participações no meio de comunicação.	
Observação:	

Figura 35: Mensagem contendo o problema, as possíveis causas e orientação

Antes, esse processo de análise dos resultados era feito pelo professor que precisava co-relacionar, manualmente, as informações do seu ambiente para chegar a uma conclusão e, assim, fazer uma orientação ao aprendiz.

7. CONSIDERAÇÕES FINAIS

No panorama atual da educação à distância, quanto ao processo de avaliação informal, os mecanismos existentes somente auxiliam o professor no acesso aos dados, consistência e qualidade.

A principal contribuição deste trabalho foi o desenvolvimento de um modelo de um mecanismo denominado MAAA para auxiliar o professor no acompanhamento do aprendiz no ambiente NetClass de ensino-aprendizagem. Este mecanismo realiza a análise das informações do aprendiz, derivando e concluindo novas informações e elabora uma orientação para o aprendiz. O professor recebe esta orientação podendo aceitar ou não ou até mesmo fazer observações. O MAAA tem as seguintes vantagens: i) arquitetura simples, de fácil entendimento e implementação; ii) o conhecimento do professor (critérios de avaliação) é separado e representado com regras de produção na sua base de conhecimento; iii) possui a capacidade de inferências.

Portanto, esse mecanismo com a capacidade de inferências baseado nos critérios de avaliação facilitou a tarefa de acompanhamento do aprendiz a qual resulta em uma tomada de decisão e, assim, diminuiu a sobrecarga de atividades no professor.

Conforme mostram as pesquisas (KEMCZINSKI, 2005; RESENDE, 2002) não existe um conjunto padrão de como o professor realiza o acompanhamento do aprendiz em um ambiente de ensino-aprendizagem a distância. Portanto, cada instituição define seu conjunto de critérios de avaliação que caracteriza como o professor acompanha e avalia um aprendiz.

O Mecanismo de Apoio no Acompanhamento do Aprendiz ainda pode ser aperfeiçoado estendendo seus critérios de avaliação, pois em sistema baseado em regras o desenvolvimento pode ser iterativo.

REFERÊNCIAS

- Almeida, M.E.B. (2003), “Educação à distância na internet: abordagens e contribuições dos ambientes digitais de aprendizagem”, *Educ. Pesqui.*, v.29, n.2, p. 327-40.
- Alves, José H. da Silva Filho. *Ontocade: um ambiente case baseado em ontologias para análise e projeto na engenharia de domínio multiagente*. Dissertação (Mestrado de Engenharia de Eletricidade) UFMA, São Luís, 2005.
- Aretio, Lorenzo Garcia; *Educación a Distancia Hoy*. Universidade Nacional de Educación a Distancia. Madrid, 1994.
- Aulanet (2000). *AulaNet 2.0 beta 3 – 1997-2000*. Fundação Padre Leonel Franca – PUC-Rio. Visitado em 25 ago. 2000. Online. Disponível na Internet: <http://anuel.cead.puc-rio.br/aulanet/index.html>.
- Barbosa, Thaisa Ferreira. *Gerenciador de Avaliações: Uma Ferramenta de Auxílio à Avaliação Formativa para o Ambiente de Educação a Distância TelEduc*. Dissertação de Mestrado. Instituto de Computação Universidade Estadual de Campinas. Campinas, 15 de dezembro de 2003.
- Bellifemine, Fabio; Caire, Giovanni; Trucco; Tiziana; Rimassa, Giovanni; MUNGENAST, Roland. *JADE Administrator's Guide*. Disponível em: <http://jade.tilab.com/doc/administratorsguide.pdf>. Acesso em: 4 nov. 2006.
- Bellifemine, Fabio; Caire, Giovanni; Trucco, Tiziana; Rimassa, Giovanni. *JADE Programmer's Guide*. Disponível em: <http://jade.tilab.com/doc/programmersguide.pdf>. Acesso em: 4 nov. 2006.
- Brookshear, J. Glenn. *Ciência da Computação: Uma Visão Abrangente*. 5ª edição. Porto Alegre: Bookman, 2000.
- Caleffe, Luis G. (2002). *A Construção do Curso em Educação a Distância - Formação de Tutores; Avaliação em Educação a Distância*. Organização de Ymiracy Nascimento S. Polak. Curitiba: Ed. do Autor.
- Charniak, Eugene; Mcdermott, Drew. *Introduction to Artificial Intelligence*. Reading MA: Addison-Wesley, 1985.
- Collis, B. (1999). *Applications of computer communications in education: an overview*. *IEEE Communications Magazine*, 82–86.
- Cruz, C. Lopes. (2003). *Midas-Poeta – Um Sistema de Apoio à Tomada de Decisão: XIV Simpósio Brasileiro de Informática na Educação – SBIE – NCE/UFRJ*.

- Demazeau, Y. From interactions to collective behaviour agent-based systems. In: Proc. of the 1st. European Conference on Cognitive Science, St. Malo, France, 1995.
- Dimitracopoulou A.; Petrou A. Advanced Collaborative Learning Systems for young students: Design issues and current trends on new cognitive and metacognitive tools.. In THEMES in Education International Journal; 60 pages, 2003.
- Evren Akar; Ebru Öztürk; Bige Tunçer e Marion Wiethoff; Evaluation of a collaborative virtual learning environment. Emerald Group Publishing Limited; Source: Education + Training, Volume 46, Numbers 6-7, 2004, pp. 343-352(10).
- Friedman-Hill, E. (2006). Jess – The rule engine for Java platform. Distributed Computing Systems. Sandia National Laboratories. Livermore, CA. Disponível em <http://herzberg.ca.sandia.gov/jess/>
- Fucks, H. Aprendizagem e trabalho cooperativo no ambiente AulaNet. Rio de Janeiro, 2000. 23p. Monografia – Faculdade de Ciência da Computação, Pontifícia Universidade Católica do Rio de Janeiro. Capturado em 12 out. 2002. Online. Disponível na Internet em: <http://piano.dsi.uminho.pt/grupok3/bibdig/aprendizagem.pdf>.
- Goldberg, M.W.; SALARI, S. An update on WebCT (world-wide-web course tools) – a tool for the creation of sophisticated Web-based learning environments. In: NAUWEB'97: CURRENT PRACTICES IN WEB BASED COURSE DEVELOPMENT, Arizona, 1997. Proceedings. Capturado em 03 dez. 2001. Online. Disponível na Internet em: <http://star.ucc.nau.edu/~nauweb97/papers/goldberg/goldberg.html>.
- Gudwin, R.R. (2004). Introdução à teoria de agentes. Disponível em <http://www.dca.fee.unicamp.br/~gudwin/courses/IA009/>. Acessado em 15 jul 2006.
- Harrer, A; McLaren, B M; Walker, E; Bollen, L; Sewall, J. Creating Cognitive Tutors for Collaborative Learning: Steps Toward Realization. User Modeling and User-Adapted Interaction. Springer-Netherlands. Volume 16, Numbers 3-4 / September, 2006.
- Hayes-Roth,F., Waterman, D. A. and Lenat, D. B.; 1978.Principles of Pattern-Directed Inference Systems. In Pattern-Directed Inference Systems, Academic Press, New York, NY, USA.
- Heejeon Suh, and Seungwook Lee. Collaborative Learning Agent for Promoting Group Interaction. ETRI Journal, vol.28, no.4, Aug. 2006, pp.461-474.

- Howe, D. (2000) Production system from FOLDOC. [online] Disponível via WWW em <http://www.nue.org/foldoc/foldoc.cgi?query=production+system>. Consultado em 17 de agosto de 2000.
- Hübner, Jomi Fred. Migração de agentes em sistemas multiagentes abertos. Dissertação (Mestrado em Ciências da Computação) UFRGS, Porto Alegre, 1995.
- Iahn, Luciana Ferreira. (2002). A Construção do Percurso em Educação a Distância – Formação de Tutores. Concepções e Políticas em Educação a Distância Organização de Ymiracy Nascimento S. Polak. Curitiba: Ed. do Autor.
- Jackson, P. (1999) Introduction to Expert Systems. 3rd Edition. London: Addison-Wesley Pub.
- JADE (2006). Disponível no endereço: <http://jade.cselt.it/>. Acesso em: 19 de Jun. 2006.
- Jaques, Patrícia Augustin. Agentes de Software para Monitoração da Colaboração em Ambientes Telemáticos de Ensino. Dissertação (Curso de Pós-Graduação em Informática). Pontifícia Universidade Católica do Rio Grande do Sul, 1999.
- Jaques, Patrícia Augustin. (2001). Agentes Pedagógicos e o Ensino Colaborativo à Distância: exame de qualificação, 143 f.
- Jaques, Patrícia Augustin; Oliveira, Flávio Moreira. (1998). Agentes de software para análise das interações em um ambiente de ensino a distância. In: III IntoEducar, Fortaleza, Brasil.
- Jaques, Patrícia Augustin; Oliveira, Flávio Moreira. (2000). Um experimento com agentes de software para monitorar a colaboração em aulas virtuais. Pontifícia Universidade Católica do Rio Grande do Sul.
- Jermann, P.; Soller, A.; Muehlenbrock, M. "From mirroring to guiding: a review of the state of the art technology or supporting collaborative learning". Proceedings of European Conference of Computer Support Collaborative Learning; Maastricht, EuroCSCL 2001.
- Jess (2006). Disponível no endereço: <http://www.jessrules.com/>. Acesso em: 06 de Jul.
- Kemczinski, Avaniilde. Método de Avaliação para Ambientes e-learning. Florianópolis, 2005. 155 f. Tese (Engenharia de Produção). Programa de Pós – Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina.
- Krammer, Érika A. W. Coester. (1990). Educação à distância: da teoria à prática. Porto Alegre: Alternativa.

- Krishnamoorthy, C. S; Rajeev, S. Artificial Intelligence and Expert Systems for Engineers; CRC Press, CRC Press LLC ISBN: 0849391253 Pub Date: 08/01/96.
- Labidi, Sofiane; Souza, C. M.; Nascimento, E. (2003). NetClass: Cooperative Learner Modeling in a Web-Based Environment In: 6th Int. Conf. On Computer Based Learning in Science. Proceedings of the 6th Int. Conf. On Computer Based Learning in Science (CBLIS) Nicosia, Cyprus: University of Cyprus.
- Larios, Edgar J; Menchaca, Rolando; Carreto, Chadwick; Flores, Augusto. Collaborative Virtual Environment (AVC) in the context of e-learning. Proceedings of the 5th IASTED international conference on web based education, 2006.
- Luger, G F. Artificial Intelligence: structures and strategies for complex solving. Addison Wesley; 4 edition (January 15, 2002).
- McCarthy, John; 1958. Programs with Common Sense. In Proceedings of the Symposium on Mechanisation of Thought Processes , 1:7784, London. Her Majesty's Stationery Office, UK.
- Mehlecke, Q T C; Tarouco, L M R. AMBIENTES DE SUPORTE PARA EDUCAÇÃO A DISTÂNCIA: A mediação para aprendizagem cooperativa. Novas Tecnologias na Educação V. 1 Nº 1, CINTED-UFRGS. 2003.
- Moore, M.; Kearsley, Greg. (1996). Study Guide for Distance Education: A Systems View. Belmont, Ca. Wadsworth Publishing Company.
- Moreira, Thiago Drumond. NetClass. (2004). Realização e implementação de um ambiente interativo de aprendizagem cooperativa assistida por computador, LSI/UFMA.
- Musa, D.; Oliveira, J.; Vicari, R. (2001). Agente para auxílio a avaliação de aprendizagem em ambientes de ensino na Web. In:Workshop de Informática na Escola.
- Newell, A.; 1982. The Knowledge Level. Artificial Intelligence 18(1):87127.
- Nilsson, Nils J. Principles of Artificial Intelligence. Morgan Kaufmann Publishers; Reprint edition, June 1986.
- Omar, Nizam; Pimentel, Edson; Mustaro, Pollyana; França, Vilma (2004) – Um Modelo para Avaliação e Acompanhamento Contínuo da Aprendizagem por parte do Professor e do Aluno – XV Simpósio Brasileiro de Informática na Educação – SBIE – Manaus.
- Otsuka, J. L; Rocha, H.V. (2002). A caminho de um modelo de apoio à avaliação contínua. In: Anais do WIE 2002, Florianópolis.

- Otsuka, J. L.; Lachi, R.; Ferreira, T.; Rocha, H. V. (2002). Suporte à Avaliação Formativa no Ambiente de Educação à Distância TelEduc . In: VI Congresso Iberoamericano de Informática Educativa, Vigo, Espanha.
- Otsuka, Joice Lee; ROCHA, Heloísa Vieira. Um Modelo de Suporte à Avaliação Formativa para Ambientes de Educação a Distância: dos conceitos à solução tecnológica, 2005.
- Placca, José Avelino. Um modelo para interações de agentes em ambientes fechados baseado em leis sociais. Niterói, RJ. Dissertação (Mestrado) – Universidade Federal Fluminense, 1999.
- Perrenoud, Philippe. Avaliação: da excelência à regulação das aprendizagens entre duas lógicas. Porto Alegre: Artmed Editora, 1999.
- Polak, Ymiracy Nascimento de S. A. (2002) Construção do Percurso em Educação a Distância – Formação de Tutores. Planejamento e Gestão de Sistemas de EAD. Organização de Ymiracy Nascimento S. Polak. Curitiba: Ed. do Autor.
- Popovic, Dobrivoje; Bhatkar, Vijay P. Methods and Tools for Applied Artificial Intelligence. Marcel Dekker, Inc. New York, NY, USA, CRC (May 2, 1994).
- Rezende, Solange Oliveira. Sistemas Inteligentes: fundamentos e aplicações. (2003) Barueri, SP: Manole.
- Riley, G. (2000) What Are Expert Systems?. [online] Disponível via WWW em <http://www.ghgcorp.com/clips/ExpertSystems.html>. Consultado em 16 de agosto de 2000.
- Romani, L. A. (2000). Intermap: Ferramenta para Visualização da Interação em Ambientes de Educação a Distância na Web. Dissertação de Mestrado, IC/Unicamp.
- Russel, S., NORVIG, P. Artificial Intelligence: A Modern Approach. New Jersey: Prentice Hall, 1995.
- Shen, R., Tang, Y.; Zhang ,T. (2001). The Intelligent Assessment System in Web_based Distance Learning Education In: 31th ASEE/IEEE Frontiers in Education Conference.
- Sichman, Jaime Simão e Alvares, Luis Otávio. Introdução aos Sistemas Multiagentes. Anais da Jornada de Atualização em Informática – JAI 97. São Paulo, 1997, editora da Sociedade Brasileira de Computação, SBC.
- Silva, D. R.; Seno, W. P.; Vieira, M. T. P. (2001). Acompanhamento do Aprendizado em Educação a Distância com Uso de Data Mining. In: Conferência Latino americana de Informática, Mérida, Venezuela.

- Silva, D. R. “Uma ferramenta para Descoberta de Conhecimento com suporte de Data Warehousing e sua aplicação para acompanhamento do aluno em Educação a Distância”. Dissertação de Mestrado, São Carlos, UFSCar, 2002, 121p.
- Silva, R. W. A. Educação a distância em ambientes de aprendizagem matemática auxiliada pela realidade virtual. 124 f. Universidade Federal de Santa Catarina. Programa de Pós-graduação. Florianópolis. 2001. Disponível em: <teses.eps.ufsc.br/defesa/pdf/7882.pdf>. Acesso em set 2006. (Dissertação de Mestrado).
- Teleduc (2000). Ambiente de suporte para ensino-aprendizagem a distância. Visitado em 29 ago. 2001. Online. Disponível na Internet em: <http://teleduc.nied.unicamp.br/teleduc/>.
- Vasconcellos, Eliane C. Vasconcellos Garcia Duarte. (2002). A Construção do Percurso em Educação a Distância: Formação de Tutores. Organização de Ymiracy Nascimento S. Polak. Curitiba: Ed. do Autor.
- Zaina, Luciana A. Martinez. “Metodologia e ferramenta para avaliação do aluno em cursos a distância via Web através de questões objetivas dinâmicas”. 107p. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo. São Paulo. 2002.
- Zaiane, O.; Luo, J. (2001). Towards Evaluating Learners’ Behaviour in a Web-Based Distance Learning Environment. In: ICALT.
- WEBCT (2000). The e-Learning hub. Visitado em 06 dez. 2001. Online. Disponível na Internet em: <http://www.webct.com>.
- WebSchool. Educação a distância. Campinas: WebSchool Educação Continuada a Distância. 2002.
- Wooldridge, M.; Jennings, N. Intelligent Agents: Theory and Practice. The Knowledge Engineering Review, [S. l.], v. 10, n. 2, p. 115-152, 1995.

APENDICE A – DEFINIÇÃO DOS COMPORTAMENTOS DO MAAA

COMPORTAMENTOS DOS AGENTES
<p>Monitor</p> <ul style="list-style-type: none"> • tipo: cíclico – jade.core.behaviours.TickerBehaviour; verifica de tempo em tempo o log; • percepção: a existência de novos fatos; • ação: envia uma mensagem para o agente conversor contendo as informações das ações e/ou atividades;
<p>Conversor</p> <ul style="list-style-type: none"> • tipo: uma única vez – jade.core.behaviours.OneShotBehaviour; • percepção: mensagem recebida do agente monitor; • ação: converte as informações do aprendiz em fatos jess;
<p>Aquisição de Conhecimento</p> <ul style="list-style-type: none"> • tipo: cíclico – jade.core.behaviours.CyclicBehaviour; fica permanentemente escutando a interface; • percepção: chamada da interface; • ação: recolhe o conhecimento do professor para convertê-lo em regras jess;
<p>Suporte ao Professor</p> <ul style="list-style-type: none"> • tipo: cíclico – jade.core.behaviours.CyclicBehaviour; • percepção: espera as mensagens dos agentes conversor e aquisição de conhecimento com os fatos e novas regras respectivamente; • ação: co-relaciona informações e deriva novas informações para elaborar a orientação

APENDICE B – CÓDIGO FONTE DOS AGENTES

```

/* Agente Monitor */
package maaa;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import java.sql.*;
import jade.acl.ACLMessage;
import maaa.*;

public class Monitor extends Agent {
    private String conceitoProva;
    private String contribuicaoChat;
    private String contribuicaoForum;
    private float acessoLicaoOnline;
    private float acessoLicaoOffline;
    private float acessoModulo;
    private Conversor conversor;
    private Conexao conexao = new Conexao();
    private ResultSet rs = null;

    protected void setup() {
        System.out.println("Agente " +getAID().getName()+ " está pronto");
        // Adiciona um comportamento TickerBehaviour que programa uma requisição a cada 5 minu
        addBehaviour( new TickerBehaviour(this, 300000) {
            protected void onTick() {
                if ( selecionaConceito() ) {
                    ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
                    msg.addReceiver(new AID("Conversor", AID.ISLOCALNAME) );
                    do {
                        atualizapriedades();
                    } while ( proximo() );
                    msg.setContent( conceitoProva + contribuicaoChat + contribuicaoForum +
                        acessoLicaoOnline + acessoLicaoOffline + acessoModulo );
                    send(msg);
                    alterarStatus();
                } else {
                    doDelete();
                }
            }
        });
    }

    //atualiza as propriedades

```

```

private void atualizaPropriedades() throws SQLException {
    conceitoProva    = rs.getString("nm_prova");
    contribuicaoChat  = rs.getString("nm_chat");
    contribuicaoForum = rs.getString("nm_forum" );
    acessoLicaoOnline = rs.getFloat("nm_licaoOff" );
    acessoLicaoOffline = rs.getFloat("nm_licaoOn");
    acessoModulo    = rs.getFloat("nm_acessoModulo");
}
/** Esse objeto passa para o proximo registro do resultset. Nao atualiza as propriedades.
 * @return A validade dessa operacao, ou seja, se há ou nao um registro
 * além do que esta atualmente o resultset do objeto
 * @throws SQLException no caso de algum erro com o banco de dados.
 */
public boolean proximo() throws SQLException {
    if ( rs.next() ){
        atualizaPropriedades();
        return true;
    }
    return false;
}
/**Mostra os conceitos novos
 * @return <code> true </code> se existir conceitos novos;
 * <code> retorna false </code> se nao conceitos novos.
 * @throws SQLException no caso de algum erro com o banco de dados.
 */
public boolean selecionaConceito( ) throws SQLException {
    rs = connectionBean.executeQuery("SELECT * FROM t_conceito WHERE flag = " + S + "");
    return this.proximo();
}
public boolean alterarStatus() throws SQLException {
    int i = connectionBean.executeUpdate("UPDATE t_conceito SET flag_isNovo = "+N+"");
    if (i > 0){
        return true;
    }
    return false;
}
}
}
/* Agente Conversor */
package maaa;
import jade.core.Agent;
import jade.core.behaviours.OneShotBehaviour;
import jade.acl.ACLmessage;;
import maaa.*;

```

```

public class Conversor extends Agent {
    protected void setup() {
        System.out.println("Oi, já estou preparado...");
        // Adiciona o comportamento que executa uma única vez;
        addBehaviour( new OneShotBehaviour() {
            public void action() {
                ACLMessage msg = myAgent.receive();
                if (msg != null) {
                    //assert uma mensagem fato em jess
                    makeassert( ACL2JessString(msg) );
                }
            }
        });
    }
    /**
     * makeasserts um fato representando uma ACLMessage em Jess.
     * chamando após a chegada da mensagem
     */
    private void makeassert(String fact) {
        try {
            jess.executeCommand(fact);
        } catch (JessException re) {
            re.printStackTrace(System.err);
        }
    }
    /**
     * @return a string representando os fatos (até mais do que um fato
     * permite, mas este metodo somente reprenta um fato a asserted em jess como)
     * uma consequencia do recepiente da messegem ACL passada
     * O conteudo da aspa antes asserting asserting o fato jess.
     * ele é unquoted pela funcao JessFact2ACL
     */
    public String ACL2JessString(ACLMessage msg) {
        String fact;

        if (msg == null) {
            return "";
        }

        // Cria uma string que asserts o fato template
        fact = "(assert (ACLMessage (communicative-act " +
            ACLMessage.getPerformative(msg.getPerformative()));
        if (msg.getSender() != null) {
            fact = fact + ") (sender " + msg.getSender().getName());
            putAIDInCache(msg.getSender());
        }
    }
}

```

```

}
Iterator i = msg.getAllReceiver();
if (i.hasNext()) {
    fact = fact + " (receiver ";
    while (i.hasNext()) {
        AID aid = (AID) i.next();
        putAIDInCache(aid);
        fact = fact + aid.getName();
    }
}
if (!isEmpty(msg.getReplyWith())) {
    fact = fact + " (reply-with " + msg.getReplyWith();
}
if (!isEmpty(msg.getInReplyTo())) {
    fact = fact + " (in-reply-to " + msg.getInReplyTo();
}
if (!isEmpty(msg.getConversationId())) {
    fact = fact + " (conversation-id " + msg.getConversationId();
}
if (!isEmpty(msg.getProtocol())) {
    fact = fact + " (protocol " + msg.getProtocol();
}
if (!isEmpty(msg.getLanguage())) {
    fact = fact + " (language " + msg.getLanguage();
}
if (!isEmpty(msg.getOntology())) {
    fact = fact + " (ontology " + msg.getOntology();
}
if (msg.getContent() != null) {
    fact = fact + " (content " + quote(msg.getContent());
}
if (!isEmpty(msg.getEncoding())) {
    fact = fact + " (encoding " + msg.getEncoding();
}
i = msg.getAllReplyTo();
if (i.hasNext()) {
    fact = fact + " (reply-to ";
    while (i.hasNext()) {
        AID aid = (AID) i.next();
        putAIDInCache(aid);
        fact = fact + aid.getName();
    }
}
if (msg.getReplyByDate() != null) {

```

```

        fact = fact + ") (reply-by " + msg.getReplyByDate().getTime());
    }
    fact = fact + ")))";
    return fact;
}
}

/* SuporteProfessor */
package maaa;
import jade.core.*;
import jade.core.behaviours.*;
import jade.lang.acl.ACLMessage;
import jess.*;
import java.io.*;
import java.util.ArrayList;
import java.util.Hashtable;
import java.util.Iterator;
import java.util.List;
import maaa.*;

public class SuporteProf extends CyclicBehaviour {

    Rete jess; //motor jess
    Agent myAgent;
    int m_maxJessPasses = 0; // assegura o numero maximo de numero de passos jess para cada execucao
    int executedPasses = -1; // conta o numero de passos jess na exec previa
    Hashtable AIDCache; // assegura cache local para mapear nomes do agente a AID

    public SuporteProf(Agent agent, String arquivoJess) {
        myAgent = agent;
        AIDCache = new Hashtable();
        // instancia do motor jess
        jess = new Rete();
        try {
            // Primeiro: definição do template ACLMessage
            jess.executeCommand( ACLJessTemplate() );
            // definir o template myagent
            jess.executeCommand("(deftemplate MyAgent (slot name))");
            // adiciona a função send()
            jess.addUserfunction(new JessSend(myAgent, this));
            // assert o fato (Myagent (name <my-name>))
            jess.executeCommand(
                "(defacts MyAgent \"All facts about this agent\" (MyAgent (name " +
                myAgent.getName() + ")))");
            // Abre o arquivo - arquivoJess

```

```

    FileReader fr = new FileReader(arquivoJess);
    // cria o parser para o arquivo, telling it where to take input
    // from and which engine to send the results to
    Jesp j = new Jesp(fr, jess);
    // parse e executa um construtor
    j.parse(false);
} catch (JessException re) {
    System.out.println(re);
} catch (FileNotFoundException e) {
    System.out.println(e);
}
}

public SuporteProf(Agent agent, String arquivoJess, int maxJessPasses) {
    this(agent, arquivoJess);
    m_maxJessPasses = maxJessPasses;
}
/**
 * Executa o comportamento
 */
public void action() {
    ACLMessage msg; // mantem a ACLMessage
    // espera uma mensagem
    if (executedPasses < m_maxJessPasses) {
        System.out.println(myAgent.getName() +
            " Estar bloqueado para esperar mensagem...");
        msg = myAgent.blockingReceive();
        // assert uma mensagem fato em jess
        makeassert(ACL2JessString(msg));
    } else {
        System.out.println(myAgent.getName() +
            " estar chegando se há mensagem...");
        msg = myAgent.receive();
        if (msg != null) {
            // assert a message fato em Jess
            makeassert(ACL2JessString(msg));
        }
    }
}
// run jess
try {
    // jess.executeCommand("(facts)");
    if (m_maxJessPasses > 0) {
        executedPasses = jess.run(m_maxJessPasses);
        System.out.println("Jess has executed " + executedPasses +

```

```

        " passes");
    } else {
        jess.run();
    }
} catch (JessException re) {
    re.printStackTrace(System.err);
}
}
private boolean isEmpty(String string) {
    return (string == null) || string.equals("");
}
/**
 * makeasserts um fato representando uma ACLMessage em Jess.
 * chamando após a chegada da mensagem
 */
private void makeassert(String fact) {
    try {
        jess.executeCommand(fact);
    } catch (JessException re) {
        re.printStackTrace(System.err);
    }
}
* Este metodo busca no cache local para AID completo do agentName passado.
* Se não encontrado ele cria AID onde somente o guid é setado.
**/
public AID getAIDFromCache(String agentName) {
    AID result;
    result = (AID) AIDCache.get(agentName);
    if (result == null) {
        result = new AID(agentName);
    }
    return result;
}
/**
 * Este metodo busca no cache local para AID completo da lista passada nomes dos agentesagent names.
 * @param context representa o contexto do motor Rete necessariamente para resolver o valor da variavel JESS
 * @param list é um ValueVector do nomes do agente
 * @return uma lista de AID
 */
public List getAIDListFromCache(Context context, ValueVector list) {
    ArrayList l = new ArrayList();
    for (int i = 0; i < list.size(); i++) {
        try {
            l.add(getAIDFromCache(list.get(i).stringValue(context)));
        }
    }
}

```

```
} catch (JessException je) {
```

APENDICE C – CÓDIGO FONTE JESS

```
(clear)
;(watch all)
;referencia para objeto elaborarOrientacao
( import maaa.ElaborarOrientacao )
( bind ?ref (new ElaborarOrientacao) )
.....
;; Templates de participacao, acesso e avaliacao
(deftemplate chat
  (slot modulo)
  (slot aprendiz)
  (multislot conceito) ;; ruim, regular, bom e excelente

(deftemplate forum
  (slot modulo)
  (slot aprendiz)
  (multislot conceito) ;; ruim, regular, bom e excelente

(deftemplate mural
  (slot modulo)
  (slot aprendiz)
  (multislot conceito) ;; ruim, regular, bom e excelente

(deftemplate acessoModulo
  (slot modulo)
  (slot aprendiz)
  (multislot conceito) ;; ruim, regular, bom e excelente

(deftemplate licaoOnline
  (slot modulo)
  (slot aprendiz)
  (multislot conceito) ;; ruim, regular, bom e excelente

(deftemplate licaoOffline
  (slot modulo)
  (slot aprendiz)
  (multislot conceito) ;; ruim, regular, bom e excelente

(deftemplate portfolio
  (slot modulo)
  (slot aprendiz)
  (multislot conceito) ;; ruim, regular, bom e excelente
```

```
(deftemplate prova
  (slot modulo)
  (slot aprendiz)
  (multislot conceito) ;; ruim, regular, bom e excelente
```

```
(deftemplate atividade
  (slot modulo)
  (slot aprendiz)
  (multislot conceito) ;; ruim, regular, bom e excelente
```

```
:: Regras
```

```
(defrule participacao
  (declare (auto-focus TRUE))
  (prova (modulo estrutura_de_dados) (aprendiz maria) (conceito ruim) )
  (forum (modulo estrutura_de_dados) (aprendiz maria) (conceito ?conceito_forum) )
  (chat (modulo estrutura_de_dados) (aprendiz maria) (conceito ?conceito_chat) )
  (acessoModulo (modulo estrutura_de_dados) (aprendiz maria) (conceito ?conceito_acessoModulo) )
  (licaoOnline (modulo estrutura_de_dados) (aprendiz maria) (conceito ?conceito_on) )
  (licaoOffline (modulo estrutura_de_dados) (aprendiz maria) (conceito ?conceito_off) )
  (portfolio (modulo estrutura_de_dados) (aprendiz maria) (conceito ?conceito_portfolio) )
  =>
  (recommend-action "prova_ruim")
  (halt)
```

```
::funcoes
```

```
(deffunction recommend-action (?conceito_prova)
  "Determina a elaboracao da orientacao"
  ( call ?ref put "conceito_prova" ?conceito_forum conceito_chat conceito_acessoModulo conceito_on conceito_off
    conceito_portfolio )
```

```
(reset)
```

```
(run)
```

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)