

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

Gustavo Pereira Mateus

**UM *FRAMEWORK* BASEADO EM SISTEMAS
MULTIAGENTES E LÓGICA *FUZZY* PARA
ACOMPANHAMENTO DE ALUNOS EM UM
AMBIENTE VIRTUAL DE ENSINO-APRENDIZAGEM**

Florianópolis

2010

Gustavo Pereira Mateus

**UM *FRAMEWORK* BASEADO EM SISTEMAS
MULTIAGENTES E LÓGICA *FUZZY* PARA
ACOMPANHAMENTO DE ALUNOS EM UM
AMBIENTE VIRTUAL DE ENSINO-APRENDIZAGEM**

Dissertação submetida ao Programa
de Pós-Graduação em Ciência da Com-
putação para a obtenção do Grau de
Mestre em Ciência da Computação.
Orientadora: Profa. Dra. Silvia Mo-
desto Nassar

Florianópolis

2010

Catálogo na fonte pela Biblioteca Universitária
da
Universidade Federal de Santa Catarina

M425f Mateus, Gustavo Pereira
Um framework baseado em Sistemas Multiagentes e Lógica Fuzzy para acompanhamento de alunos em um ambiente virtual de ensino-aprendizagem [dissertação] / Gustavo Pereira Mateus ; orientadora, Silvia Modesto Nassar. - Florianópolis, SC, 2010. 79 p.: il., grafs.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.

Inclui referências

1. Ciência da computação. 2. Lógica Fuzzy. 3. Ambiente virtual - Estudo e ensino - Avaliação. 4. Ensino a distância. I. Nassar, Silvia Modesto. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU 681

Gustavo Pereira Mateus

**UM *FRAMEWORK* BASEADO EM SISTEMAS
MULTIAGENTES E LÓGICA *FUZZY* PARA
ACOMPANHAMENTO DE ALUNOS EM UM
AMBIENTE VIRTUAL DE ENSINO-APRENDIZAGEM**

Esta Dissertação foi julgada aprovada para a obtenção do Título de “Mestre em Ciência da Computação”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 29 de novembro 2010.

Prof. Dr. Mario Antonio Ribeiro Dantas
Coordenador do Curso

Profa. Dra. Silvia Modesto Nassar
Orientadora

Banca Examinadora:

Profa. Dra. Rosa Maria Vicari

Profa. Dra. Araci Hack Catapan

Prof. Dr. Ricardo Azambuja Silveira

AGRADECIMENTOS

À minha orientadora Silvia pelo apoio e dedicação. Seus conselhos foram fundamentais para a realização deste trabalho.

Ao meu colega Vilson pelas dicas e sugestões. Sua cooperação foi importante para que esta pesquisa se tornasse realidade.

Aos professores Araci, Rosa e Ricardo por aceitarem o convite para participar da banca e por suas contribuições.

Agradeço com muito carinho à Beatriz pela ajuda, compreensão e paciência durante todos os momentos que passamos juntos. Sua força de vontade e perseverança foram exemplos essenciais em minha vida.

Aos meus familiares pelos momentos que não pude compartilhar.

Agradeço, também, a todos que contribuíram com o desenvolvimento desta dissertação, oferecendo incentivo para o andamento da pesquisa.

RESUMO

Essa pesquisa tem o objetivo de promover o acompanhamento do aluno por meio da análise do desempenho dele em um Ambiente Virtual de Ensino-Aprendizagem (AVEA). Dessa forma é proposto um *framework* que integra a arquitetura de agentes *Belief-Desire-Intention* (BDI) com a Lógica *Fuzzy* (LF). Assim, são considerados quatro agentes: Estudante, Avaliação, Atividade de aprendizagem e Acompanhamento. O agente Estudante é um agente cognitivo, que tem as crenças modeladas pela teoria dos conjuntos *fuzzy*. Os demais agentes são agentes reativos. Os agentes Avaliação e Atividades de aprendizagem coletam dados do ambiente e trocam informações com o agente Estudante. Esse agente processa os dados e determina o desafio pedagógico adequado ao aluno. Na sequência, o agente Acompanhamento é acionado e realiza a ação pedagógica determinada. Essa arquitetura foi implementada em um AVEA que é utilizado por alunos de graduação da UFSC. Foram utilizadas as linguagens de programação PHP, *Perl* e C, bem como o banco de dados MySQL. Observou-se, por meio de avaliações com os alunos, que o *framework* proposto tem uma perspectiva positiva para integração em AVEA. Sendo possível realizar um acompanhamento contínuo do desempenho do aluno e estimulá-lo a uma utilização mais apropriada dos recursos de aprendizagem.

Palavras-chave: Agentes. SMA. Avaliação. Lógica *Fuzzy*. Ambiente Virtual de Ensino-Aprendizagem. Educação a Distância.

ABSTRACT

This research aims to provide the monitoring of students through the analysis of his performance in a Virtual Learning Environment (VLE). Thus a framework, that involves a Belief-Desire-Intention (BDI) agent architecture and Fuzzy Logic (FL), is proposed. This framework is composed of four agents: student, evaluation, learning activities and monitoring. The student agent is a BDI agent, which has beliefs modeled by the fuzzy set theory. The other agents are reactive agents. The learning activities and evaluation agents collect data from the environment and exchange information with the student agent. This agent processes the data and determines the appropriate educational stimulus according to the student model. Following, the monitoring agent is triggered to perform the given pedagogical action. This architecture was implemented in a VLE that is used by graduate students at UFSC. The programming languages PHP, Perl and C, and the MySQL database were used in this work. It was observed, by means of evaluations with the students, that the proposed framework has a positive outlook for integration into VLE. Thus it is possible, through continuous monitoring of student's performance, to encourage them to use more appropriate learning resources.

Keywords: Agents. MAS. Evaluation. Fuzzy Logic. VLE. Distance Teaching.

LISTA DE FIGURAS

Figura 1	Um agente no ambiente	26
Figura 2	Arquitetura para um agente cognitivo	28
Figura 3	Abstração de um modelo BDI	31
Figura 4	Sistemas Multiagentes	33
Figura 5	Principais funções de pertinência	36
Figura 6	Sistema de Lógica <i>Fuzzy</i>	37
Figura 7	Métodos de desfuzzyficação	39
Figura 8	Diagrama de objetivos do sistema	45
Figura 9	<i>Framework</i> proposto	46
Figura 10	Agentes envolvidos	47
Figura 11	Visão geral do sistema	47
Figura 12	Agente BDI	48
Figura 13	Especificação do agente estudante	49
Figura 14	Representação da biblioteca de níveis <i>fuzzy</i>	49
Figura 15	Módulo tutoria	51
Figura 16	Módulo avançado	52
Figura 17	Módulo auto-avaliação	53
Figura 18	Modelagem das tabelas para registro dos <i>logs</i>	54
Figura 19	Mapeamento dos agentes do <i>framework</i> para o AVEA ..	55
Figura 20	Protótipo da arquitetura do SMA	56
Figura 21	Comunicação entre os agentes	57
Figura 22	Regras de inferência	59
Figura 23	Nível <i>fuzzy</i> 1: Maior incerteza	60
Figura 24	Nível <i>fuzzy</i> 2: Incerteza moderada	61
Figura 25	Nível <i>fuzzy</i> 3: Incerteza pequena	61
Figura 26	Nível <i>fuzzy</i> 4: Menor incerteza	61
Figura 27	Visualização do agente de interface	62
Figura 28	Tabela com o desempenho dos alunos	63
Figura 29	Acompanhamento dos alunos	64
Figura 30	Análise do acompanhamento dos alunos	64
Figura 31	Distribuição dos alunos nos níveis	65

LISTA DE ABREVIATURAS E SIGLAS

EaD	Educação a Distância	17
AVA	Ambiente Virtual de Aprendizagem	17
AVEA	Ambiente Virtual de Ensino-Aprendizagem	17
TCD	Tecnologia de Comunicação Digital	17
SMA	Sistema Multiagente	18
BDI	<i>Belief-Desire-Intention</i>	18
LF	Lógica <i>Fuzzy</i>	18
IA	Inteligência Artificial	22
RS	Estado de raciocínio	28
DS	Estado de decisão	28
CS	Estado de engajamento	28
SLF	Sistema de Lógica <i>Fuzzy</i>	37
FIS	<i>Fuzzy Inference System</i>	37
MATLAB	<i>MATrix LABORatory</i>	38
AIA	Ambientes Inteligentes de Aprendizagem	40
OIA	Objetos Inteligentes de Aprendizagem	41
PDT	<i>Prometheus Design Tool</i>	44
PHP	<i>Hypertext Preprocessor</i>	53
SQL	<i>Structured Query Language</i>	53
<i>Perl</i>	<i>Practical Extraction and Report Language</i>	53
BD	Banco de Dados	53
MER	Modelagem Entidade Relacionamento	54

SUMÁRIO

1 INTRODUÇÃO	17
1.1 OBJETIVOS	20
1.1.1 Objetivo Geral	20
1.1.2 Objetivos Específicos	20
1.2 PROCEDIMENTOS METODOLÓGICOS	20
1.3 ORGANIZAÇÃO DO TRABALHO	22
2 FUNDAMENTAÇÃO TEÓRICA	25
2.1 AGENTE	25
2.1.1 Arquiteturas reativas	27
2.1.2 Arquiteturas cognitivas	28
2.1.2.1 Modelo de agentes BDI	29
2.1.3 Arquiteturas de agentes de interface	31
2.2 SISTEMAS MULTIAGENTES	32
2.3 FUNDAMENTOS DA LÓGICA <i>FUZZY</i>	34
2.3.1 Funções de pertinência	35
2.3.2 Inferência em sistemas <i>fuzzy</i>	37
2.3.3 O processo de desfuzzyficação	38
2.4 TRABALHOS RELACIONADOS	39
3 O <i>FRAMEWORK</i> PROPOSTO	43
3.1 DEFINIÇÃO DOS AGENTES	43
3.1.1 Especificação do sistema	44
3.1.2 Arquitetura do projeto	45
3.1.3 Detalhamento do projeto	48
4 IMPLEMENTAÇÃO DO <i>FRAMEWORK</i>	51
4.1 O AMBIENTE UTILIZADO COMO ESTUDO DE CASO ..	51
4.2 EXTENSÃO DO BANCO DE DADOS DO AVEA	53
4.3 AGENTES NO AVEA	54
4.3.1 A comunicação dos agentes	56
4.3.2 Crenças, Desejos e Intensões	57
4.4 MOTOR DE INFERÊNCIA <i>FUZZY</i>	58
4.4.1 Modelagem <i>fuzzy</i>	59
4.4.1.1 Biblioteca de níveis	60
4.5 AGENTE DE INTERFACE	62
4.6 RESULTADOS	62
5 CONCLUSÕES	67
REFERÊNCIAS	69
APÊNDICE A – Artigos Publicados	77

1 INTRODUÇÃO

Segundo (SOUZA, 2007), a Educação a Distância (EaD) é uma forma de organização de ensino-aprendizagem na qual alunos estudam em grupo ou individualmente em seus lares, locais de trabalho ou outros lugares com materiais auto instrutivos distribuídos por meios de comunicação, possibilitando a comunicação com docentes, tutores ou outros alunos.

A maioria dos professores utiliza ambientes virtuais para disponibilizar informações e materiais *online*. Porém, esses ambientes possuem pouco suporte para avaliar e discriminar os diferentes comportamentos dos alunos no ambiente virtual de ensino, bem como a forma de execução das atividades *online* propostas, durante a realização dos cursos (ZAIANE; LUO, 2001). Nesta pesquisa, propõe-se a utilização de agentes de *software* para coletar informações do aluno e consequentemente gerar desafios pedagógicos de acordo com a análise do desempenho dos alunos.

De acordo com Roncarelli e Catapan (2009), Ambiente Virtual de Aprendizagem (AVA) e Ambiente Virtual de Ensino-Aprendizagem (AVEA) têm naturezas distintas que não se esclarecem somente pela descrição das funcionalidades operacionais, tecnológicas ou ergonômicas da vasta quantidade de ambientes virtuais atualmente disponíveis. O AVA é qualquer ambiente mediado pela Tecnologia de Comunicação Digital (TCD) e que é amplamente explorado pelas comunidades do ciberespaço. E um AVEA constitui-se em um processo definidamente sistemático, organizado, intencional e de caráter formal.

Normalmente os professores elaboram um relatório de acompanhamento de avaliação, que é feito para acompanhar o andamento da aprendizagem dos alunos. Com ele, tem-se uma visão geral do andamento da turma em relação às avaliações e atividades que são propostas. Neste trabalho gera-se esse relatório para que o professor possa acompanhar o andamento da aprendizagem dos alunos por meio das atividades cadastradas (listas de exercícios e provas). Este módulo, que gera o relatório, objetiva a visualização de um “*ranking*” dos alunos por turma facilitando o planejamento do tempo necessário para que os alunos concluam as atividades e identifica grupos de alunos mais adiantados.

Assim, a comunidade de agentes, proposta nesta pesquisa, atua em um AVEA para coletar informações e identificar qual é a avaliação geral do aluno, em um determinado momento, no ambiente.

Por definição, um agente é uma entidade computacional com um

comportamento autônomo que lhe permite decidir suas próprias ações (ALVARES; SICHMAN, 1997). A decisão de qual ação será executada é determinada pelo agente, tendo em consideração as mudanças acontecidas no ambiente em que atua e o desejo de alcançar seus objetivos. A ideia principal em um Sistema Multiagente (SMA) é que um comportamento global inteligente pode ser alcançado a partir do comportamento individual dos agentes. Em um SMA não é necessário que cada agente seja individualmente inteligente para alcançar um comportamento global inteligente.

Este trabalho é construído a partir da proposta de SMA e está direcionado para o modelo de agentes *Belief-Desire-Intention* (BDI). Os conceitos deste modelo foram inicialmente propostos por Bratman (1987). O modelo consiste de crenças, desejos e intenções. Isso envolve as atitudes mentais que geram a ação humana. Rao e Georgeff (1995) adaptaram o modelo proposto por Bratman (1987), transformando-o em uma teoria formal e um modelo de execução para agentes de *software* baseados na noção de crenças, objetivos e planos.

Implementar agentes BDI em um AVEA envolve a definição de quais serão as crenças, desejos e intenções dos agentes dentro do ambiente. Esta pesquisa propõe que as crenças representem o nível de conhecimento que o agente presume que o aluno tenha sobre um conteúdo específico. Para isso foi necessário integrar a arquitetura de agentes BDI um sistema de Lógica *Fuzzy* (LF), pois existe um grau de incerteza envolvido no processo de definição do desempenho do aluno. As intenções, um subgrupo dos desejos, são estimular e ajudar o aluno no processo de aprendizagem, por meio de mensagens de apoio e dicas de utilização do ambiente. Essas mensagens são fornecidas por meio de um agente de interface, o qual tem como proposta apoiar e conduzir os alunos a utilizar os recursos mais adequados e que estão disponíveis no ambiente, conforme sua aprendizagem.

Segundo Reategui et al. (2006), SMA são utilizados como uma alternativa eficiente na modelagem de ambientes interativos. Além disso, segundo Oliveira, Vicari e Ceron (1996), os princípios dos SMA mostram-se bastante adequados ao desenvolvimento de ambientes computacionais de ensino, tendo em vista que o problema de ensino-aprendizagem é de natureza social e cooperativa.

Neste trabalho é desenvolvido um *framework* para acompanhamento do desempenho dos alunos em um AVEA. A proposta desta pesquisa é que sejam implementados diferentes desafios pedagógicos para acompanhar o processo de aprendizagem dos alunos. Para o desenvolvimento desse *framework* foram utilizadas técnicas de modelagem e

implementação de agentes, bem como a aplicação de conjuntos difusos.

Uma das possibilidades de criar um modelo de adaptação para AVEA baseado em SMA envolve selecionar as características que são implementadas na maioria dos ambientes. Nessa perspectiva, este trabalho busca o desenvolvimento de uma proposta que possa consolidar as características gerais que são necessárias para a implementação dos agentes. Tais características envolvem, por exemplo, o registro de *logs* e avaliações do desempenho de alunos através de tarefas como os questionários.

Com esses atributos é possível criar uma comunidade de agentes que trabalhem com essas características e identifiquem o desempenho do aluno. Além disso, a comunicação entre os agentes pode, a todo momento, adaptar a estratégia pedagógica de acordo com o desempenho identificado.

Neste sentido, é necessário analisar os dados obtidos a partir de agentes do AVEA, e aplicar LF ou Lógica Difusa, para descobrir informações relevantes sobre o desempenho do aluno com relação as atividades desenvolvidas. Essas informações são extraídas do banco de dados do ambiente e são utilizadas para a classificação do desempenho do aluno em regular, bom e excelente. De acordo com Jameson (1995), a LF exige poucos cálculos e é mais fácil de inicializar e atualizar o modelo do aluno. A LF se baseia em regras que contém variáveis linguísticas, o que torna mais intuitiva a modelagem dos sistemas (PEDRYCZ, 2004).

Uma das principais potencialidades da LF, quando comparada com outros esquemas que tratam com dados imprecisos como redes neurais, é que suas bases de conhecimento, as quais estão no formato de regras de produção, são fáceis de examinar e entender. Este formato de regra também torna fácil a manutenção e a atualização da base de conhecimento.

As avaliações são atividades imprescindíveis no processo de ensino-aprendizagem, pois indicam a evolução ou o nível de compreensão do aluno sobre o conteúdo. Elas consistem no julgamento por pessoas (professores/tutores) ou sistemas sobre o desempenho de outras pessoas (alunos). Segundo Rasmani (2002), as formas tradicionais de avaliação, baseadas em métodos aritméticos e estatísticos são utilizadas pela grande maioria das instituições educacionais. A LF é uma alternativa para a avaliação qualitativa por utilizar-se de conceitos linguísticos (RIBEIRO, 2007).

Desta forma, nesta pesquisa, procura-se acompanhar e identificar diferentes comportamentos de aprendizagem através da LF. O que pode promover uma aprendizagem colaborativa, já que é possível rela-

cionar alunos com o mesmo desempenho em determinados tópicos da disciplina. Além disso, são gerados relatórios com o aproveitamento geral das turmas no sistema.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo desta pesquisa é desenvolver um *framework* para integrar a arquitetura de agentes BDI com LF, que possa ser aplicado em diversos AVEA, para acompanhar e avaliar o desempenho de alunos durante a realização de tarefas disponibilizadas no AVEA.

1.1.2 Objetivos Específicos

- Identificar e selecionar os recursos de AVEA que permitem avaliar o aluno.
- Modelar a sociedade de agentes que compõem o *framework*.
- Modelar o agente que permite identificar o desempenho do aluno.
- Modelar as crenças do agente BDI por meio de LF.
- Prover desafios pedagógicos a partir da identificação do desempenho.
- Validar o *framework* proposto para acompanhar o desempenho dos alunos.

1.2 PROCEDIMENTOS METODOLÓGICOS

Os procedimentos metodológicos desta pesquisa envolvem a determinação e identificação dos recursos que são considerados pertinentes para analisar e capturar o desempenho dos alunos. Com esse foco foram realizadas análises sobre amostras de utilização de um AVEA. Foi utilizada uma amostra dos alunos matriculados na disciplina de Estatística do segundo semestre de 2009. O objetivo dessa análise foi levantar quais são os recursos do AVEA que são considerados relevantes para extrair os dados sobre a avaliação do desempenho do aluno.

Dessa forma, vários recursos do AVEA que armazenam dados dos alunos no servidor foram analisados. Foram observados recursos como o tempo de sessão nas páginas visitadas pelos alunos, o desempenho deles em questionários avaliativos, o desempenho em atividades práticas, o tempo total utilizado para responder o questionário entre outras informações. Assim foram estudadas as variáveis que seriam significativas para construção dos agentes e entrada do modelo *fuzzy* para avaliação do desempenho.

Em princípio o tempo nas sessões de aprendizagem seria avaliado com uma variável linguística para o sistema *fuzzy*. No entanto, depois de realizadas algumas análises foi constatado que não havia correlação entre tempo no AVEA e o desempenho do aluno. Foi observado que alunos com um tempo extremamente alto poderiam evidenciar não só o tempo de estudo, mas sim o tempo em outras atividades externas ao AVEA.

Segundo Roncarelli e Catapan (2009), um AVEA precisa atender a todo o processo educativo, em suas questões legais, funcionais, acadêmicas e pedagógicas. Portanto, em qualquer AVEA são considerados enfoques práticos e teóricos. Respectivamente denominados, neste trabalho, de avaliação prática e avaliação teórica. Assim, nessa pesquisa foi considerado um mapeamento dessas diferentes avaliações de desempenho do aluno no intuito de encontrar uma analogia, que poderia ser contemplada no modelo *fuzzy* proposto.

Segundo Bardin (2002) é um equívoco privilegiar o conhecimento imediato (empírico) em detrimento daquele fundamentado em conceitos, na falsa ideia de que a teoria (uma abstração) é desprovida de objetividade. Inexiste prática sem teoria, bem como teoria desprovida de prática. Sendo assim, o foco desta pesquisa concentrou-se tanto na avaliação teórica como a avaliação prática justificando e preservando o movimento contínuo de teoria-prática.

Considerou-se relevante, também, perceber o movimento de aprendizagem. O movimento de aprendizagem é definido como o processo onde o aluno compreende os conceitos e põem em prática a teoria assimilada. São os avanços, retrocessos e tomadas de decisões que são realizadas nas páginas do ambiente. Nesse trabalho houve a preocupação em tratar somente os caminhos de aprendizagem que fossem considerados mais seguros para as análises. Isso quer dizer que só seriam considerados os caminhos que fossem contínuos, sem idas e voltas. As idas e voltas podem caracterizar um aluno não tão seguro quanto a sua análise dos dados e tomada de decisão. No ambiente, esse tipo de comportamento de ida e volta é bem mais observado nas fases iniciais

de estudo, o que denota uma incerteza sobre qual o caminho é mais adequado seguir.

Desta forma, para acompanhar e reconhecer o nível de aprendizagem em AVEA foram definidas as variáveis (recursos) que fornecem o desempenho do aluno, considerando avaliações na parte teórica e na parte prática. Assim, o *framework* proposto abrangeu a definição de quatro agentes: o agente “estudante”, agente “avaliação”, agente “atividade de aprendizagem” e o agente de “acompanhamento”. Os agentes “avaliação” e “atividade de aprendizagem” são agentes que coletam os dados relativos aos desempenhos dos alunos, considerando a parte teórica e prática respectivamente. O agente “estudante” é o agente que faz o raciocínio sobre a avaliação do desempenho dos alunos através de um motor de inferência de LF. E o agente “acompanhamento” é o responsável por executar o desafio pedagógico definido pelo agente “estudante” através da avaliação da nota, que foi obtida pelo motor de inferência.

A validação do *framework* proposto consistiu na implementação dos agentes de *software* para monitorar e extrair informações de utilização do ambiente pelos alunos. A validação e implementação do *framework* foi realizada em um AVEA que é utilizado por alunos da graduação da UFSC.

1.3 ORGANIZAÇÃO DO TRABALHO

Após o capítulo introdutório é apresentada a fundamentação teórica desta pesquisa. Nesse capítulo 2 são abordados os aspectos relacionados as técnicas de Inteligência Artificial (IA) aplicadas em diversos sistemas e ambientes interativos. São detalhados os agentes, arquiteturas reativas e cognitivas, agentes de interface e SMA. Na sequência são apresentados fundamentos e conceitos sobre LF. Também são relatados modelos cognitivos de alunos em ambientes, além das perspectivas e trabalhos relacionados com está pesquisa.

No capítulo 3 é apresentada a proposta do *framework* envolvendo as definições, a modelagem dos agentes reativos e cognitivos, bem como a integração da LF com o agente cognitivo. No capítulo 4 são apresentados aspectos relacionados com a implementação e uso dos agentes envolvidos na arquitetura de um AVEA. Também é apresentada e detalhada a modelagem de LF que é aplicada sobre os dados extraídos pelos agentes.

Por fim, no capítulo 5, são apresentadas as conclusões desta pes-

quisa, bem como resultados e observações realizadas sobre a implementação do *framework* em um AVEA. Além disso, é apresentada a proposta de continuação e os trabalhos futuros desta pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentadas as definições e conceitos empregados nesta pesquisa. Também são abordados alguns trabalhos relacionados a esta linha de pesquisa.

Em diversos trabalhos como de Giraffa (1999) e Vicari (1988), já era mencionado o uso de técnicas de IA com o objetivo de desenvolver sistemas computacionais de ensino. Sistemas com capacidade de adaptação ao contexto e personalização do ambiente de acordo com as características dos alunos. Além disso, nesses sistemas gera-se um alto grau de interatividade entre o ambiente e os usuários com um controle maior das sessões de aprendizagem em ambientes multiusuários.

2.1 AGENTE

Existem várias definições sobre o termo agente e isso ocorre porque determinadas características de um agente têm a sua relevância atrelada ao contexto de sua aplicação. Pesquisadores da comunidade de IA têm dificuldade em chegar a uma definição universal. Parte dessa dificuldade é devida à várias propriedades de agentes terem diferentes importâncias em diferentes domínios.

De acordo com Brenner, Rüdiger e Wittig (1998), um agente deve possuir um certo grau de inteligência para executar as suas tarefas. É esse nível de inteligência que permite a um agente agir de maneira autônoma. Além disso, para este autor, um agente deve interagir com o seu ambiente.

Para Russell e Norvig (2004) um agente é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores. Além disso, pode-se distinguir agentes racionais e agentes irracionais que, dentro da Inteligência Artificial, são denominados como agentes cognitivos e reativos, respectivamente. Os agentes reativos diferenciam-se dos agentes cognitivos pelo fato de não agirem deliberadamente e sim devido a eventos externos, ou seja, apenas reagem a situações, por isso o nome reativo. Por essa razão um agente reativo é mais simples que um agente cognitivo e por si só não realiza modificações significativas no ambiente onde atua.

A definição de Wooldridge (2000) para agente é que ele é um sistema de computador que é situado em algum ambiente, e que é

capaz de realizar ações autônomas neste, para alcançar seus objetivos de projeto. A Figura 1 mostra uma visão abstrata de um agente, onde podemos ver as saídas de ações geradas.



Figura 1 – Um agente no ambiente
Fonte: Adaptado de Wooldridge (2009)

Ainda de acordo com Wooldridge (2009) na maioria de domínios de complexidade razoável, um agente não tem controle total sobre seu ambiente. Ele pode ter o melhor controle parcial em que ele pode influenciar. Do ponto de vista do agente a mesma ação realizada duas vezes em circunstâncias aparentemente idênticas pode parecer ter efeitos totalmente diferentes, e ainda pode falhar em ter os efeitos desejados. Assim, agentes devem ser preparados para a possibilidade de falhar. Pode-se dizer que, em geral, ambientes são assumidos como não determinísticos (estocásticos/não totalmente previsíveis/imprevisíveis).

Segundo Wooldridge (2009) normalmente um agente terá um repertório de ações disponíveis, que define a capacidade do agente de modificar seu ambiente. Nem todas as ações podem ser realizadas em todas as situações. Portanto, ações têm pré-condições associadas a elas, que definem as possíveis situações onde elas podem ser aplicadas.

De um modo geral um agente de *software* é conceituado como uma entidade que funciona de forma contínua e autônoma em um ambiente em particular, geralmente habitado por outros agentes, e que seja capaz de intervir no seu ambiente, de forma flexível e inteligente, sem requerer intervenção ou orientação humana constante. E de um modo ideal, um agente que funcione, continuamente, por longos períodos de tempo, deve ser capaz de aprender com a experiência e, se ele habita um ambiente com outros agentes, seja capaz de comunicar-se e cooperar com eles.

Um agente realiza o mapeamento de uma percepção em uma ação conforme a sua arquitetura. A arquitetura de um agente específica como este pode ser decomposto em um conjunto de módulos e como estes módulos devem interagir. O conjunto de módulos e suas interações

descrevem como os dados recebidos do ambiente e o estado interno do agente determinam suas ações (WOOLDRIDGE, 2000).

Segundo Sichman (1995) as arquiteturas dos agentes dividem-se em arquiteturas cognitivas, as quais apresentam agentes complexos, e arquiteturas reativas, associadas a agentes mais simples. Existe, também, uma mescla entre essas arquiteturas denominada de arquitetura híbrida. Além disso, surgiram outras denominações como os agentes de *software* e os agentes de interface. Nas subseções a seguir são detalhadas essas arquiteturas.

2.1.1 Arquiteturas reativas

De acordo com Oliveira, Vicari e Ceron (1996), as arquiteturas reativas são próprias de agentes que possuem estrutura interna simples e interação de forma limitada; geralmente não possuem representação dos estados mentais; o desempenho da sociedade é resultado do número de agentes e da rapidez nas interações.

Para Brooks (1986), um agente reativo possui módulos de competência, que permitem que o mesmo responda a eventos particulares do seu ambiente. Os sensores percebem o ambiente e encaminham as informações para os módulos de competência, os quais realizam o processamento que irá gerar a reação adequada.

O modelo de funcionamento de um agente reativo é formado pelo par estímulo-resposta (ação-reação). As principais características dos agentes reativos são destacadas a seguir (FERBER; GASSER, 1991):

- não há representação explícita do conhecimento: o conhecimento dos agentes é implícito (as suas regras de comportamento) e sua manifestação se externa através do seu comportamento e dos demais agentes;
- não há representação do ambiente: o comportamento (resposta) de cada agente é baseado no que ele percebe (estímulo) a cada instante. Não há uma representação interna explícita do ambiente;
- não há memória das ações: os agentes reativos não mantêm nenhum tipo de histórico de suas ações, ou seja, o resultado de uma determinada ação passada não influencia diretamente na decisão de uma ação futura.

Em geral, os agentes reativos são muito simples e não possuem

representação do seu ambiente. Suas reações dependem unicamente de sua percepção deste ambiente. Alguns agentes do SMA proposto neste trabalho são implementados de acordo com a definição da arquitetura de agentes puramente reativos.

2.1.2 Arquiteturas cognitivas

Conforme Oliveira, Vicari e Ceron (1996), as arquiteturas cognitivas são associadas a agentes complexos, com mecanismos de inferência e decisão robustos, interações sofisticadas e alto grau de intencionalidade no comportamento. Na arquitetura definida (Fig. 2) em Sichman (1995) visualiza-se a estrutura interna de um agente cognitivo como sendo composto por mecanismos de percepção e recepção de mensagens (entrada de dados), ação e emissão de mensagens (saída de dados), raciocínio e decisão (estados internos) e revisão das crenças do agente. O agente raciocina socialmente sobre os demais agentes. Tal mecanismo utiliza as informações que o agente tem sobre os demais, armazenadas numa estrutura denominada descrição externa. O agente também possui diversos estados internos, como o estado de raciocínio (RS), de decisão (DS) e de engajamento (CS). Tais estados são modificados através da execução dos diversos mecanismos internos.

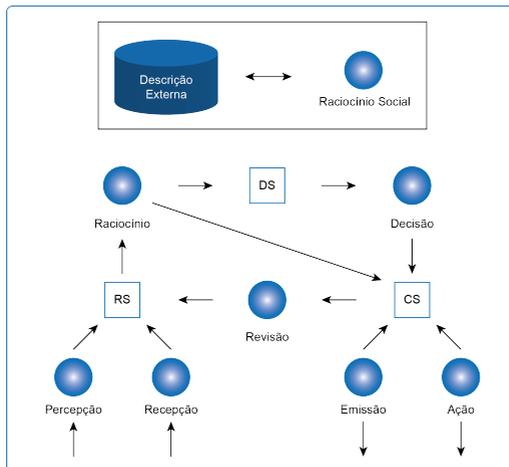


Figura 2 – Arquitetura para um agente cognitivo

Fonte: Sichman (1995)

Segundo Oliveira, Vicari e Ceron (1996), as arquiteturas cognitivas dividem-se em: Arquiteturas funcionais e arquiteturas baseadas em estados mentais. Nas arquiteturas funcionais o agente é dividido em módulos que implementam as funcionalidades consideradas necessárias a sua operação.

Ainda de acordo com Oliveira, Vicari e Ceron (1996) as arquiteturas baseadas em estados mentais adotam uma perspectiva psicológica para definição da estrutura de agentes. Os componentes mentais: crença, capacidades, escolha e compromisso, devem ser definidos de forma precisa e ter uma correspondência direta com seu uso no senso comum.

Nesse trabalho foi implementado um agente cognitivo baseado em estados mentais. A próxima seção apresenta o modelo BDI que representa uma arquitetura cognitiva, e tem sua origem no modelo de raciocínio humano.

2.1.2.1 Modelo de agentes BDI

A arquitetura de agentes BDI é um modelo de *software* desenvolvido para programar agentes inteligentes. Desenvolver agentes BDI com o intuito de modelar o conhecimento baseado em estados mentais, semelhante ao raciocínio humano, tem sido investido em várias pesquisas.

De acordo com Wooldridge (2000), o modelo BDI combina três componentes distintos que compreendem o componente filosófico, componente de arquitetura de *software* e componente lógico.

O componente filosófico define que o modelo BDI é fundamentado na teoria de ação racional em humanos, proposta pelo filósofo Bratman (1987). O componente de arquitetura de *software* define que o modelo de agentes BDI não prescreve uma implementação específica, ou seja, o modelo pode ser implementado de maneiras diferentes.

De acordo com Fagundes (2004) esses três componentes tornam o modelo BDI um dos mais conhecidos e estudados modelos de raciocínio prático existentes. As ações do agente são derivadas a partir desse processo de raciocínio prático, o qual é constituído de dois passos. No primeiro passo, deliberação de objetivos, faz-se a seleção de um conjunto de desejos que devem ser alcançados, de acordo com a situação atual das crenças do agente. O segundo passo, raciocínio meio-e-fim, é responsável pela determinação de como esses desejos concretos, produzidos como resultado do passo anterior, podem ser atingidos através

do uso dos meios disponíveis ao agente.

A ideia central de adotar os estados mentais do modelo BDI se concentra no fato de que o agente cognitivo possui estados internos que se relacionam com estado do ambiente com o qual interage. Estes estados são correspondentes aos estados mentais humanos: crenças, desejos e intenções.

Segundo Georgeff et al. (1999), as crenças representam o conhecimento sobre o mundo. Em termos computacionais, crenças são apenas uma maneira de representar o estado do mundo, seja através de variáveis, uma base de dados relacional, ou expressões simbólicas em um cálculo de predicados.

Para Giraffa (1999), os desejos são relacionados eventualmente ao estado de mundo que o agente pretende provocar. Os desejos não dirigem necessariamente o agente a agir, isto é, o fato de um agente possuir um desejo não significa agir para o satisfazer. Significa que antes de um determinado agente decidir o que fazer, ele passa por um processo de racionalização e confronta os seus desejos com as suas crenças. O agente escolherá os desejos que são possíveis seguindo algum critério.

De acordo com Fagundes (2004), as intenções correspondem aos estados de mundo que o agente quer efetivamente provocar, ou seja, existe um comprometimento em realizá-las. Podem ser consideradas um subconjunto dos desejos, mas ao contrário destes, devem ser consistentes. As intenções são formadas a partir de um processo de deliberação e a partir do refinamento de outras intenções. No entanto, um agente pode conter intenções iniciais inseridas pelo usuário. A Figura 3 ilustra a abstração de um modelo BDI.

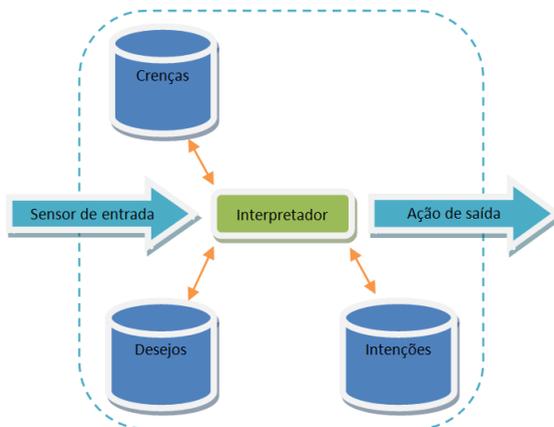


Figura 3 – Abstração de um modelo BDI

2.1.3 Arquiteturas de agentes de interface

Segundo Costa (1999), o agente de interface interage com o usuário e coopera com os outros agentes trocando informações sobre capacidades, comprometimentos e metas de aprendizado dos usuários.

Estes componentes de *software* podem ou não ter uma forma de visualização gráfica. Quando isso ocorre, são normalmente chamados de agentes de interface animados, ou agentes de interface personificados (RABELLO; REATEGUI; COSTI, 2007).

De acordo com Rabello, Reategui e Costi (2007) os agentes de interface normalmente têm a função de dar apoio à utilização de uma aplicação em particular. Os agentes de interface possuem papéis importantes de conselheiro, guia e facilitador. Mas para que estas funções sejam efetivadas, outra característica é fundamental para dar credibilidade ao usuário, que é a presença de afetividade no personagem adotado.

Neste trabalho é desenvolvido um agente de interface que é representado por um personagem animado que envia mensagens verbais aos alunos envolvidos no ambiente. Esse agente é inicializado a partir de um agente cognitivo implementado na arquitetura deste trabalho.

2.2 SISTEMAS MULTIAGENTES

Os SMA representam uma subárea da IA e concentram-se no estudo de agentes autônomos em um universo multiagente. Para os SMA, o termo autônomo designa o fato de que os agentes têm uma existência própria, independente da existência de outros agentes. Usualmente, cada agente possui um conjunto de capacidades comportamentais que definem sua competência, um conjunto de objetivos, e a autonomia necessária para utilizar suas capacidades comportamentais a fim de alcançar seus objetivos. Um agente é uma entidade computacional com um comportamento autônomo que lhe permite decidir suas próprias ações (ALVARES; SICHMAN, 1997). A ideia principal em um SMA é que um comportamento global inteligente pode ser alcançado a partir do comportamento individual dos agentes. Em um SMA não é necessário que cada agente seja individualmente inteligente para alcançar um comportamento global inteligente.

De acordo com Jennings, Sycara e Wooldridge (1998) um SMA pode ser definido como uma rede de resolvidores de problema que trabalham juntos para solucionar problemas que estão além das capacidades individuais (ou conhecimentos individuais) de cada resolvidor de problema. Ou seja, quando a resolução de um objetivo necessita o esforço de dois ou mais agentes, denominamos de um SMA.

Um SMA pode ser visto como uma rede, fracamente acoplada, de solucionadores de problemas que trabalham em conjunto para resolver problemas que vão além da sua capacidade individual. Estes solucionadores de problemas são essencialmente autônomos, distribuídos e, muitas vezes, heterogêneos em sua natureza. A Figura 4 mostra um ambiente multiagente onde cada agente tem influência em apenas uma parte do ambiente e precisam interagir (GUEDES, 2005).

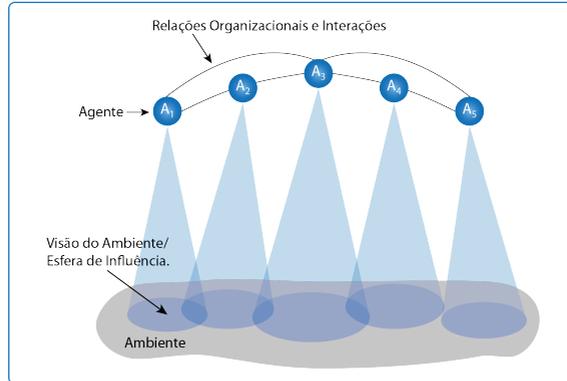


Figura 4 – Sistemas Multiagentes
 Fonte: Adaptado de Guedes (2005)

Denomina-se interação entre agentes ou entre agente/ambiente uma troca de informações, que pode ser realizada de forma direta (comunicação explícita) ou de modo indireto (emissão de sinais através do ambiente). Uma organização define todas as restrições aplicadas aos agentes pertencentes a uma determinada sociedade, ou seja, os meios através dos quais o projetista do sistema pode garantir que cada agente desejará e realizará a resolução dos problemas propostos.

Os SMA podem ser caracterizados didaticamente em duas classes. A primeira denomina-se SMA reativos que trabalham com o desenvolvimento de sistemas que utilizam um grande número de agentes simples para a resolução de um determinado problema. A segunda abordagem, denominada SMA cognitivos trabalha com poucos agentes que realizam tarefas mais complexas que os primeiros.

Moulin e Chaib-Draa (1996) enfatizam as características que constituem vantagens significativas dos SMA sobre o solucionador de problemas monolítico, dentre elas:

- Maior rapidez na resolução de problemas através do aproveitamento do paralelismo;
- Diminuição da comunicação, por transmitir somente soluções parciais em alto nível para outros agentes, ao invés de dados brutos para algum lugar central;
- Mais flexibilidade, por ter agentes com diferentes habilidades que são dinamicamente agrupados para resolver problemas.

De acordo com Guedes (2005) ambientes colaborativos de ensino, que estão sendo usados como uma ferramenta para a viabilização da EaD, possuem uma natureza dinâmica e descentralizada. É dinâmica porque os alunos estão em constante aprendizado mudando o ambiente de ensino e descentralizada pela própria natureza, os alunos, na maioria das vezes, estarão geograficamente separados. Para a resolução de problemas distribuídos os conceitos de SMA podem ser bem aplicados. Nestes problemas há a necessidade de interação entre vários agentes devido a diversos fatores como o não conhecimento de todo o contexto por parte de um único agente.

2.3 FUNDAMENTOS DA LÓGICA *FUZZY*

A lógica *fuzzy* tenta aproximar a decisão computacional da decisão humana, tornando as máquinas mais autônomas. Isto é feito de forma que a decisão de uma máquina não se resume apenas a um “sim” ou um “não”, mas também tenha decisões “talvez”. É um modo de interligar inerentemente processos analógicos que deslocam-se através de uma faixa contínua para um computador digital que trabalha com valores numéricos bem definidos (valores discretos).

Trabalhar com fatores como ambiguidade, incerteza e informações vagas na resolução de problemas é uma característica do pensamento humano, que usa o conhecimento adquirido e experiências para lidar com esses fatores (KLIR; CLAIR; YUAN, 1997). Segundo Mukaidono (2001) nos problemas de difícil solução, em que se faz necessário o auxílio matemático/computacional, modelar tais fatores é extremamente difícil. A modelagem computacional convencional não trabalha com ambiguidades, pois utiliza o conceito de verdadeiro ou falso. Segundo Klir, Clair e Yuan (1997) A precisão computacional também é limitada, deixando sempre uma margem, por mínima que seja, para a incerteza. A vagueza de informações advinda da linguagem natural agrava ainda mais os fatores citados. Para lidar com isso de forma matemática, foi desenvolvida por Zadeh e Kacprzyk (1992) a teoria dos conjuntos *fuzzy* (difusos), teoria a qual permite serem tratados níveis de incerteza e ambiguidade.

De acordo com Adilea (2003), ao contrário da lógica tradicional, LF não impõe limites bruscos, proporcionando graus de pertinência de elementos a uma determinada categoria. Pode-se definir lógica difusa como sendo uma ferramenta capaz de capturar informações vagas, em geral descritas em linguagem natural e convertê-las para um formato

numérico, de fácil manipulação. A representação destas informações vagas se dá através do uso de conjuntos difusos (*Fuzzy Set*).

Para Santos (2003), quando nos deparamos com um problema em uma determinada área, normalmente, deve-se, de início, pesquisar e estudar os processos que ocorrem nesse domínio e, em seguida, transformá-lo em um modelo matemático que o simule. Quando estuda-se sistemas complexos, tais procedimentos tornam-se uma tarefa muito difícil, já que há a necessidade de se conhecer detalhadamente todo o processo que se deseja controlar e que o mesmo esteja bem definido.

Uma área de aplicação da LF é chamada raciocínio aproximado, conforme Zadeh e Kacprzyk (1992), onde um tipo de raciocínio não é totalmente certo nem totalmente errado. Este tipo de raciocínio se aproxima da forma do raciocínio humano. Neste caso as variáveis linguísticas são representadas por conjuntos *fuzzy*, interpretando uma variável linguística como uma variável cujos valores são palavras ou sentenças em uma linguagem natural. Conjuntos *fuzzy* proporcionam, aos métodos de desenvolvimento de sistemas computacionais, uma forma de programação mais próxima da linguagem e do raciocínio humano.

De acordo com Klir e Yuan (1995), a lógica clássica, ainda é utilizada em larga escala resolvendo muitos dos problemas encontrados em variadas áreas. Entretanto, há situações de problemas reais em que é impossível modelar matematicamente determinados processos. É o caso de sistemas que apresentam uma gama muito grande de informações e que as características desses dados são conhecidos apenas linguisticamente. Assim, a utilização da lógica binária leva a respostas imprecisas e inexatas, invalidando sua utilização.

A utilização da LF, a qual trabalha com informações e termos linguísticos, pode solucionar este tipo de problema. A modelagem e controle *fuzzy* são técnicas que conseguem trabalhar rigorosamente essas informações analisando não apenas a exatidão dos resultados, mas também o grau de incerteza.

2.3.1 Funções de pertinência

Para Klir, Clair e Yuan (1997) cada conjunto *fuzzy*, A , é definido em termos de relevância a um conjunto universal por uma função denominada de função de pertinência, associando a cada elemento x um número, $A(x)$, no intervalo fechado $[0, 1]$ que caracteriza o grau de pertinência de x em A . A função de pertinência tem a forma: $\mu_A : x \rightarrow [0, 1]$

Ainda segundo Pedrycz e Gomide (1998), as principais funções de pertinência utilizadas são a forma triangular(a), função-G(b), função-S(c), trapezoidal(d), gaussiana(e) e exponencial(f). Na Figura 5 são apresentadas as representações de cada uma destas, que são os padrões mais comuns de funções de pertinência.

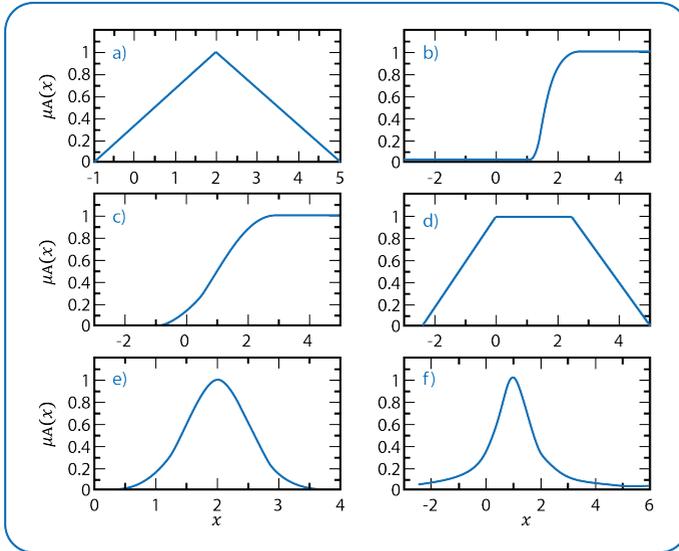


Figura 5 – Principais funções de pertinência
Fonte: Costa (2006)

Cada curva representa um conjunto ou subconjunto *fuzzy*. O eixo das abscissas representa um número pertencente a este conjunto e no eixo das ordenadas o grau de pertinência deste ao conjunto. A notação do grau de pertinência é representada da seguinte forma $\mu_A(x) = 0,70$. Isto diz que o grau de pertinência do elemento x ao conjunto A é de 0,70.

Como na teoria tradicional de conjuntos, os conjuntos *fuzzy* também podem ser manipulados algebricamente com operações de união, interseção e complemento, contudo estas operações são definidas em termos do grau de pertinência dos conjuntos.

Considerando dois conjuntos *fuzzy*, A e B , e um elemento x com grau de pertinência $\mu_A(x)$ e $\mu_B(x)$, sua união, interseção e o complemento de A podem ser representados pelas operações padrão:

$$\begin{aligned}\mu_{A \cup B}(x) &= \max[\mu_A(x), \mu_B(x)] \\ \mu_{A \cap B}(x) &= \min[\mu_A(x), \mu_B(x)] \\ \mu_{\bar{A}}(x) &= 1 - \mu_A(x)\end{aligned}$$

2.3.2 Inferência em sistemas *fuzzy*

De acordo com Mendel (2001), uma abordagem livre de modelos que é baseada em regras e LF leva ao que é chamado de Sistema de Lógica *Fuzzy* (SLF). Ele define o SLF como um sistema que faz um mapeamento não-linear de um vetor de entrada em uma saída escalar. Basicamente a LF é usada quando se tem um sistema complexo com pequena quantidade de dados e informação imprecisa, que é o caso de modelos baseados em conhecimento especialista. A Figura 6 é uma representação de SLF.

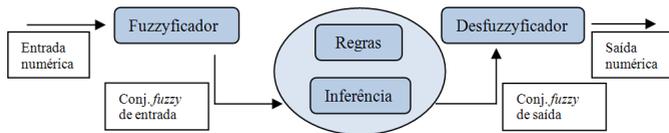


Figura 6 – Sistema de Lógica *Fuzzy*

No fuzzyficador ocorre o mapeamento da entrada rígida (*crisp*) em um número *fuzzy*. As regras são o centro do sistema, é onde está a base de conhecimento utilizada pelo sistema de inferência *fuzzy*. Elas podem ser obtidas através de conhecimento especialista ou extraídas de dados numéricos e são expressas na forma de estruturas “SE-ENTÃO”. A parte da condição, o “SE”, é chamado de antecedente e o resultado, o “ENTÃO”, é o consequente. Aqui entram as funções de pertinência, pois os conjuntos *fuzzy* são associados aos termos do antecedente, consequente, entradas e saídas do sistema.

O sistema de inferência *Fuzzy Inference System* (FIS) mapeia conjuntos *fuzzy* de entrada para conjuntos *fuzzy* de saída. Ele faz a manipulação da ativação e combinação das regras através de operações entre duas funções de pertinência. Existem vários procedimentos de inferência, mas os mais comuns são o mínimo e o produto escalar.

O processo de inferência *fuzzy* envolve todas as etapas de composição das funções, operações lógicas, e as regras “SE-ENTÃO”. No MATLAB (*MATrix LABoratory*) (MATLAB, 2010) é possível implementar dois tipos de métodos de inferência *fuzzy*: *Mamdani* e *Sugeno*. Estes

dois tipos de sistemas de inferência variam um pouco na forma como as saídas são determinadas.

O controlador de *Mamdani* tem como base o trabalho pioneiro de *Mamdani* (MAMDANI, 1973). No algoritmo *fuzzy* desse controlador, cada regra é uma proposição condicional *fuzzy*. A implementação de cada regra é feita mediante a definição de operadores para o processamento do antecedente da regra e da função de implicação que irá definir o seu conseqüente. A ação do controlador *fuzzy* é definida pela agregação das regras que compõem o algoritmo, mediante o uso de um conectivo que pode ser implementado por diferentes operadores. Esta agregação resulta em um conjunto *fuzzy*, que define a saída do controlador. A saída efetiva do controlador é obtida por meio de um processo de desfuzzyficação aplicado ao conjunto.

Já o controlador de *Sugeno* (TAKAGI; SUGENO, 1983) é uma simplificação do controlador de *Mamdani*, onde o conseqüente de cada regra é definido como uma função das variáveis linguísticas de entrada. O resultado de cada regra é um valor numérico (não um conjunto *fuzzy*). Assim a função de implicação não é necessária. A resposta final do controlador é obtida pela média ponderada das respostas das regras individuais.

2.3.3 O processo de desfuzzyficação

O desfuzzyficador simplesmente faz a redução do conjunto *fuzzy*, resultante da inferência, em um número *crisp*. O termo desfuzzyficador equivale à transformação *fuzzy* \rightarrow escalar, com o objetivo de produzir um valor numérico. Assim variáveis linguísticas (conjuntos), produzidas pela máquina de inferência, devem ser desfuzzyficadas. A seguir são apresentados alguns métodos para desfuzzyficação (SHAW; SIMÕES, 1999):

- Maior dos máximos: O valor da variável de saída corresponde ao maior índice com pertinência máxima ao conjunto *fuzzy* de saída;
- Menor dos máximos: O valor da variável de saída corresponde ao menor índice com pertinência máxima ao conjunto *fuzzy* de saída;
- Média dos máximos: O valor numérico da saída corresponde ao índice referente a média dos máximos locais do resultado da agregação das regras de inferência;

- Centro de gravidade: é o método mais utilizado, se baseia no cálculo do centro de gravidade da função de agregação. Calcula-se a área do conjunto *fuzzy* de saída e acha-se o índice correspondente ao centro de gravidade.

A Figura 7 mostra alguns métodos de desfuzzyficação e os respectivos valores de saída de um FIS com uma variável de entrada x , uma variável de saída y e duas regras de inferência. As funções de pertinência para a variável x são representadas por A e B , assim como C e D representam as funções de pertinência da variável y .

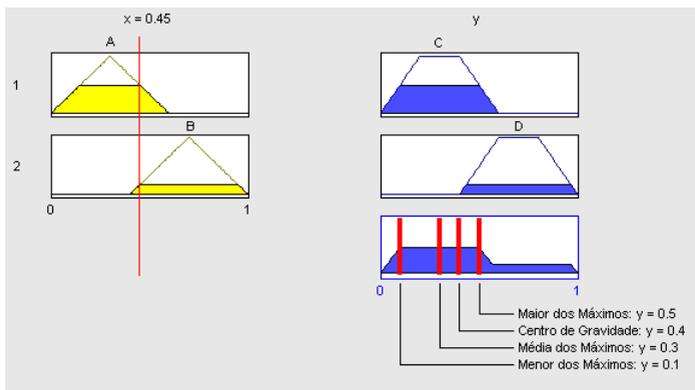


Figura 7 – Métodos de desfuzzyficação

2.4 TRABALHOS RELACIONADOS

Um ponto central e de grande interesse de diversas pesquisas tem sido a possibilidade de construir AVEA baseados em SMA para garantir suporte ao aprendizado e adaptação às experiências requisitadas pelos alunos. Muitas pesquisas têm a preocupação de minimizar as tarefas do professor na identificação das dificuldades de aprendizagem dos alunos.

Em trabalhos como de Piva et al. (2005) já se mencionava a implementação de um mecanismo que, além de facilitar o processo de avaliação dos alunos, procurava por casos similares ocorridos com outros alunos em situações passadas semelhantes. Esse mecanismo permitia que o sistema redefinisse o conteúdo a que o aluno era submetido, para eliminar as deficiências detectadas durante a avaliação formativa, sem a necessidade de uma maior intervenção do professor.

Nesta pesquisa também se tem a intenção de gerar resultados ao professor no intuito de beneficiar o processo de ensino-aprendizagem como um todo. Para isso, o SMA implementado carrega as informações do desempenho do aluno e o direciona ao módulo de aprendizagem mais conveniente. Todo esse movimento é registrado e pode ser verificado pelo professor para que o mesmo possa acompanhar a turma de um modo geral. Além disso, o comportamento do sistema é adaptado ao desempenho do aluno de acordo a aplicação de LF. De modo que, dinamicamente, o aluno seja orientado a trabalhar nas atividades mais adequadas ao seu processo de aprendizagem.

Muitas pesquisas propõem métodos, ferramentas e ambientes de acompanhamento de desempenho baseados em conjuntos *fuzzy* (por exemplo, satisfatório, insatisfatório e regular), onde o professor relaciona o grau de pertinência desses valores dentro de intervalos de notas, assim como a base de regras (Nolan (1998); Weon e Kim (2001); Fabri e Fabri (2002); Rasmani (2002); Hadjileontiadou et al. (2004); Rieder e Brancher (2004); Rissoli e Giraffa (2006); Stathacopoulou et al. (2005); Huang et al. (2007); Ribeiro (2007); Silva et al. (2008)).

Na pesquisa de Vahldick, Santiago e Raabe (2008) já eram exemplificados os usos da LF em Ambientes Inteligentes de Aprendizagem (AIA), tendo sido um forte auxílio a adoção dessa técnica de IA.

A vantagem na utilização da LF é considerar a avaliação individual como dependente do desempenho geral, e assim permitir que se experimentem ajustes das configurações (pertinência e base de regras) para normalizar a nota individual seguindo a média da turma. Por exemplo, mesmo que a média de aprovação seja 6,0 (seis), tais sistemas conseguem apontar que um aluno com média 5,5 (cinco e meio) pode ser considerado aprovado. Nesse processo de ajuste, o valor numérico que representa a média do aluno ainda pode ser recalculado para gera a nota final. Alguns desses trabalhos ajustam ou propõem automaticamente o grau de pertinência para refletir o histórico das turmas anteriores, e ainda conseguem predizer o desempenho individual, alertando tanto o professor quanto o aluno baseando-se em suas notas atuais e o histórico das turmas.

Esses trabalhos citaram as seguintes entradas para o sistema *fuzzy*: taxa de acertos e erros em testes, quantidade de acessos e tempo na leitura do material instrucional, tempo para execução dos testes (por questão e total), tentativas em acertar os exercícios, assiduidade em acessar o ambiente, nível de dificuldade, complexidade e importância das questões. Como saídas dos sistemas pode-se enumerar: aprovação, empenho, previsão de aprovação, recomendações para o aluno (aces-

sar determinado material, fazer exercícios, participar das atividades em grupo), habilidade, motivação, estilo de aprendizado, grau de colaboração e proveito nas atividades.

Na pesquisa de Bavaresco, Silva e Silveira (2008) é implementado um modelo de agentes BDI para criação de Objetos Inteligentes de Aprendizagem (OIA) com o objetivo de gerar resultados de acompanhamento para o professor. Isto porque os OIA têm a capacidade de fornecer informações do aluno para o professor. Entende-se que um OIA é uma entidade que consiste de um agente capaz de desempenhar o papel de um objeto de aprendizagem (SILVEIRA; GOMES; VICARI, 2006). No trabalho de Fagundes (2007) os estados mentais de agentes BDI foram representados por um modelo probabilístico que trata a incerteza da informação.

Além disso, o desenvolvimento desta pesquisa busca, pela implementação de agentes BDI, a possibilidade de modelar o conhecimento baseado em estados mentais. O que seria semelhante ao raciocínio humano, que tem sido investido em várias pesquisas.

3 O *FRAMEWORK* PROPOSTO

A proposta dessa pesquisa envolve a modelagem de uma comunidade de agentes que possa ser adaptada a qualquer AVEA. O objetivo é promover um acompanhamento da aprendizagem dos alunos por meio da análise do desempenho individual durante um curso em um AVEA. Nessa perspectiva, propõe-se a implementação de uma comunidade de agentes que implementem diferentes funcionalidades para atingir o objetivo proposto. Dessa forma pretende-se utilizar agentes reativos, agentes cognitivos BDI e agentes de interface.

A ideia de uma população de agentes surge com o objetivo de criar, na proposta desta pesquisa, uma abordagem que contemple os principais tipos de agentes, com comportamentos e objetivos bem definidos, para que o *framework* de SMA proposto possa ser adaptado a qualquer AVEA. A tarefa de projetar um SMA que possa analisar o desempenho dos alunos de modo individual e fazer inferências sobre essas análises não é uma atividade trivial. Essa modelagem requer uma análise detalhada sobre as tarefas e objetivos envolvidos no SMA.

Um segundo conceito chave, dentro dessa proposta, é a estratégia adotada por estes agentes. A estratégia envolve a maneira como um agente reage ao seu meio e persegue seus objetivos. No intuito de atender as estratégias de adaptação do *framework* proposto e considerando a incerteza envolvida para atingir o objetivo desta pesquisa, propõe-se a integração das técnicas de LF ao agente BDI. Essa integração é atendida modelando cada uma das crenças deste agente utilizando diversas funções de pertinência, selecionadas dinamicamente. Desse modo, é possível observar os aspectos de adaptação vinculados às estratégias do agente. Isso ocorre no intuito de melhorar o processo em que o agente está envolvido, a fim de alcançar com maior eficiência seus objetivos. Pois as funções podem ser modificadas de acordo com os resultados esperados.

3.1 DEFINIÇÃO DOS AGENTES

Para facilitar a análise e especificação dos agentes envolvidos no *framework* do SMA proposto, foi utilizada a metodologia do Prometheus (2010). A implementação e validação dessa proposta em um AVEA será apresentada no capítulo seguinte.

Prometheus (PADGHAM; WINIKOFF, 2002) é uma metodologia

para desenvolvimento de SMA que abrange desde a modelagem até a sua implementação. Esta metodologia é composta por três fases, onde os artefatos produzidos são utilizados tanto na geração do esqueleto do código, como também para a depuração e o teste. Sua aplicação é possível através da ferramenta *Prometheus Design Tool* (PDT) que suporta a construção dos artefatos seguindo a proposta da metodologia.

A primeira fase correspondente a especificação do sistema, é composta por duas etapas: determinar o ambiente do sistema (percepções e ações) e determinar os objetivos e funcionalidades do sistema (objetivos e cenários de casos de uso). A fase dois é o projeto arquitetural, que utiliza as saídas obtidas na fase anterior para determinar quais agentes irão compor o sistema e como os mesmos irão interagir. Esta fase envolve três etapas: a definição dos tipos de agentes, a definição da estrutura do sistema e a definição das interações entre os agentes. Por fim a última fase é o projeto detalhado, que define a capacidade dos agentes, eventos internos, planos e uma estrutura de dados detalhada por agente.

3.1.1 Especificação do sistema

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

De acordo com a Figura 8, o diagrama de objetivos deste trabalho possui três papéis principais: “acompanhar a evolução do desempenho”, “determinar o desafio pedagógico” e “interagir com o aluno”. O papel de acompanhar a evolução do desempenho relaciona-se com os objetivos de coletar dados, ou seja, extrair os resultados das avaliações e atividades de aprendizagem.

Para determinar o desafio pedagógico é necessário utilizar um agente cognitivo, pois neste ponto ocorre a análise dos dados e a tomada de decisão. Assim, o agente pode selecionar o desafio pedagógico mais adequado ao desempenho do aluno. O conjunto de desafios é a lista de desejos disponíveis para o agente, portanto a intenção do agente é executar um dos desafios pedagógicos selecionados (Fig. 8). Essa ação deve ser processada por um agente BDI, com mecanismos de inferência e tomada de decisões.

O acompanhamento do aluno é representado pelo papel de “in-

teragir com aluno”. Nessa etapa são enviados desafios pedagógicos ao aluno. A determinação de qual é o desafio apropriado, conforme já mencionado, é definida na etapa anterior e envolve a implementação de mecanismos de inferência. A Figura 8 mostra o diagrama de objetivos gerado pelo PDT.



Figura 8 – Diagrama de objetivos do sistema

3.1.2 Arquitetura do projeto

A arquitetura do projeto define e especifica os diferentes agentes envolvidos nesta proposta. Dessa forma são propostos quatro agentes: um agente BDI que é denominado agente “estudante”, dois agentes reativos denominados “avaliação” e “atividade de aprendizagem” e um agente de interface denominado de agente de “acompanhamento”. A Figura 9 apresenta a ideia do *framework* proposto.

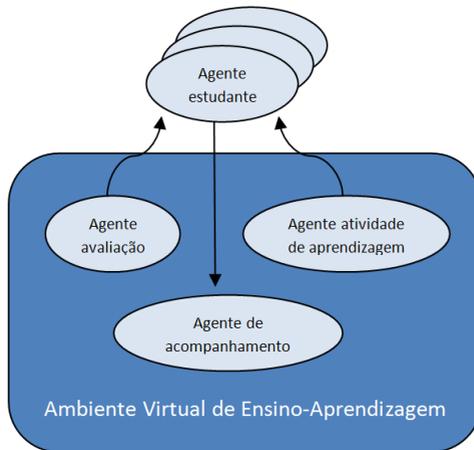


Figura 9 – *Framework* proposto

A definição dos agentes envolvidos baseou-se nas características gerais dos atuais AVEA. Isso significa que qualquer AVEA possui recursos ou módulos que possam avaliar o aluno e disponibilizar atividades de aprendizagem. Na arquitetura proposta o agente “estudante” é uma representação (instância) do aluno ativo no AVEA. O agente “avaliação” pode se basear em questionários e exercícios de avaliação no ambiente. As atividades de aprendizagem são consideradas por meio de participação em *wikis* e fóruns, que não necessariamente envolvem uma nota ou conceito. O agente de “acompanhamento” é o agente responsável pela comunicação com o aluno, enviando desafios pedagógicos recomendados ao seu desempenho.

De acordo com Weiss (1999), em SMA os agentes se comunicam e interagem para alcançarem objetivos comuns e individuais. Na proposta apresentada os agentes de “atividade de aprendizagem” e “avaliação” têm o mesmo foco: acompanhar a evolução do desempenho. A parte que envolve determinar o desafio pedagógico é realizada pelo agente “estudante”, de acordo com o acompanhamento da evolução do desempenho do aluno. O agente “acompanhamento” é o que efetua a ação de interagir com o aluno. A Figura 10 mostra o diagrama de papéis e agentes gerado no PDT.

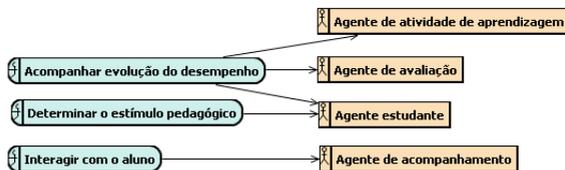


Figura 10 – Agentes envolvidos

As percepções são as diferentes informações que podem ser coletadas do AVEA. O agente “estudante” processa as percepções através de um motor de inferência *fuzzy* com de uma biblioteca de níveis pré-determinada, e envia o tipo de desafio adequado para o agente “acompanhamento”, que efetiva a ação determinada. Assim, o aluno é conduzido a utilizar outros desafios pedagógicos, pode ser informado sobre quais são os colegas que estão na mesma fase (desempenho similar), pode ser sugerido que estude mais os conceitos base da disciplina ou qualquer desafio definido na implementação. A Figura 11 mostra a relação entre os agentes do sistema, as percepções do ambiente (desempenhos), algumas ações que podem ser tomadas pelo agente “estudante” (desafios) e exemplos de mensagens que podem ser enviadas ao aluno pelo agente “acompanhamento”.

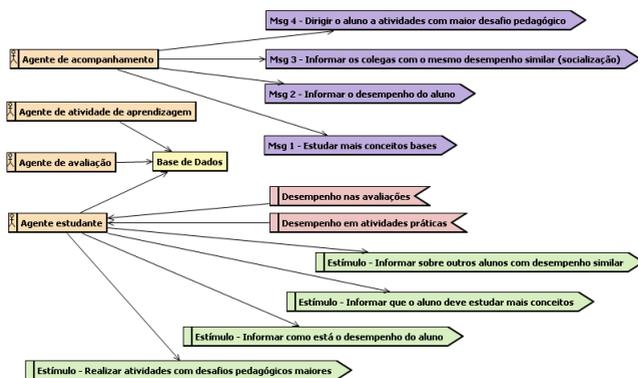


Figura 11 – Visão geral do sistema

3.1.3 Detalhamento do projeto

O detalhamento do projeto envolve a especificação dos agentes envolvidos. Como os agentes “avaliação” e “atividade de aprendizagem” são agentes reativos, que rastreiam e armazenam dados dos alunos, e o agente “acompanhamento”, também reativo, só executa a ação determinada, não é necessário apresentar uma especificação detalhada dos mesmos utilizando a ferramenta PDT. Nessa etapa de detalhamento do projeto, considerou-se conveniente apresentar o agente cognitivo “estudante”.

A aplicação do modelo BDI, nessa pesquisa, abrange a definição das crenças, desejos e intenções do agente “estudante”. O agente “estudante” é uma representação de um agente BDI por possuir estados mentais. As crenças desse agente envolvem as definições do que o sistema calcula que o aluno sabe sobre uma determinada parte do conteúdo. Essa avaliação, por tratar com incerteza, é determinada por um conjunto de funções de pertinência que é denominada, neste trabalho, como uma biblioteca de níveis *fuzzy*. Os desejos são os estados que o agente “estudante” pretende provocar no aluno. Conforme definição a base de intenções é um subconjunto dos desejos do agente “estudante”, ou seja, os estados que efetivamente serão provocados. A definição de qual intenção será utilizada pelo agente “estudante” para estimular o aluno envolve um processo de racionalização. Nesse processo o agente “estudante” confronta seus desejos com suas crenças. A Figura 12 representa o processo de racionalização do agente BDI “estudante” nesse contexto.

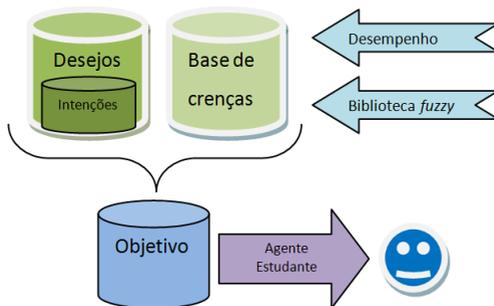


Figura 12 – Agente BDI

O agente “estudante” possui vários níveis definidos. Cada nível é especificado por diferentes funções de pertinência para as variáveis linguísticas (ex.: regular, bom e excelente) que recebem os desempenhos do aluno. A ideia de implementar os níveis por meio da LF é justificada pela incerteza presente na determinação de como está o desempenho geral do aluno. É através disso que se determina o desafio pedagógico mais adequado ao aluno, que é processado pelo agente de interface “acompanhamento”. A especificação da Figura 13 é do agente “estudante”, o agente cognitivo BDI proposto neste *framework*.

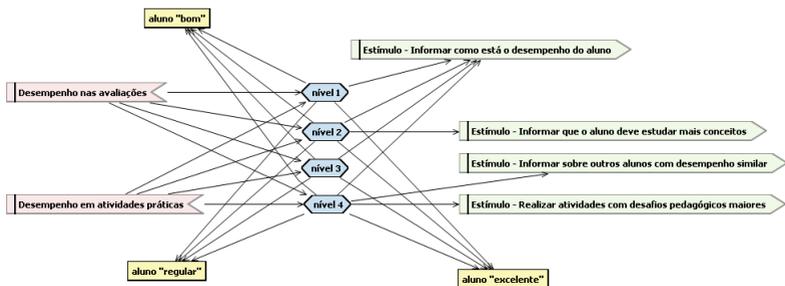


Figura 13 – Especificação do agente estudante

Os níveis, da biblioteca *fuzzy*, representam o grau de incerteza que o agente tem sobre o conhecimento do aluno, iniciando da maior incerteza (N_1) e progredindo para a menor incerteza (N_n). Para avançar de nível o aluno deve manter a nota de saída do sistema *fuzzy* acima de um percentual estabelecido em cada nível, caso contrário, ele retrocede no conjunto de níveis da biblioteca *fuzzy*. A Figura 14 mostra como são definidos os níveis da biblioteca e os tipos de desafios gerados.

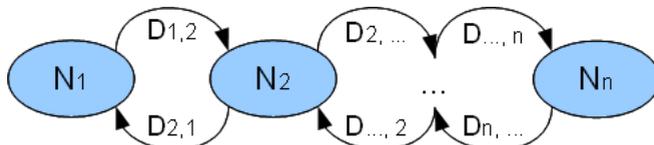


Figura 14 – Representação da biblioteca de níveis *fuzzy*

Os desafios são ativados no intervalo entre o avanço ou retrocesso do aluno nos níveis. Assim são definidos desafios que variam de acordo

com a mudança de nível, podendo existir desafios no avanço ($D_{1,2}$ a $D_{2,n}$) ou no retrocesso ($D_{n,2}$ a $D_{2,1}$) dos níveis.

4 IMPLEMENTAÇÃO DO *FRAMEWORK*

4.1 O AMBIENTE UTILIZADO COMO ESTUDO DE CASO

O ambiente utilizado como estudo de caso chama-se SestatNet (OLIVEIRA et al., 2006), ele é usado para ensino semipresencial na disciplina de Estatística pelos alunos de engenharia da Universidade Federal de Santa Catarina. No ambiente, o aluno além de aprender sobre diversos conceitos sobre Estatística é, também, conduzido a projetar uma pesquisa quantitativa. Nessa pesquisa, os dados coletados são estruturados em uma planilha que é importada para o ambiente. O aluno tem a possibilidade de trabalhar os dados de sua pesquisa em dois módulos: tutoria e avançado.

Inicialmente o aluno trabalha no módulo de tutoria, e lá aprende o conteúdo estatístico guiado pelo Mapa Conceitual (MC). Cada passo do aluno está vinculado a uma planilha de dados, que pode ser de sua pesquisa ou um exemplo disponibilizado no AVEA. A Figura 15, mostra a interface do módulo tutoria do AVEA utilizado nesta pesquisa.

The screenshot displays the SestatNet web application interface. The main content area shows a question about the normality test for a quantitative variable. The question asks to verify the normality of a variable 'TAA' and to choose the correct hypothesis (H0 or H1) based on the test results. The interface includes a conceptual map on the left, a search system at the top right, and a test configuration panel at the bottom. The test configuration panel shows the following details:

- Teste de Aderência à Curva de Distribuição Normal**
- Nome do Arquivo: `QUOT_t_de_student1.csv`
- Variável Dependente: `TAA` (with a green checkmark)
- Variável Independente: `trabalha` (with a green checkmark)
- Eliminar os Valores Discrepantes?: Não Sim
- Status: A variável dependente 'TAA' está mensurada como 'OT Contínua'.
- Status: A variável independente 'trabalha' está mensurada como 'OL Nominal'.
- Status: O nível de significância é de '0,05'.
- Teste de Aderência Qui-quadrado: Normalidade
- Resultado: $\chi^2 = 1,1$ e $p = 0,90$

At the bottom of the configuration panel, there is a table with the following structure:

Classes	Freq. Observada	P(Oi)	E(Oi)

On the right side of the interface, there is a histogram showing the distribution of the data, with a normal distribution curve overlaid. The histogram has a title 'Normalidade' and a y-axis labeled 'Frequência'.

Figura 15 – Módulo tutoria

O MC é um mapa que contém os principais tópicos e respectivos

subtópicos abordados na disciplina. Nesse módulo de tutoria o aluno tem total liberdade para escolher seus caminhos de aprendizagem. O MC mostra ao aluno, em formato de Árvore de Decisão, o caminho de navegação que uma dada interação percorre até chegar ao resultado estatístico final, e também os caminhos que o sistema pode seguir no caso de respostas diferentes. Essa característica permite localizar o aluno dentro do raciocínio estatístico.

No módulo avançado do AVEA o aluno aplica diretamente, sobre seus dados, a análise desejada sem o apoio do módulo de tutoria, ou seja, sem o uso do MC. Normalmente este módulo é usado pelos alunos depois que eles se sentem seguros sobre as análises que desejam realizar, assim que eles tenham aprendido como se faz uma análise de dados no módulo tutoria. A Figura 16, apresenta a interface do módulo avançado.

The screenshot shows the SestatNet web application interface. The browser title is "SEstatNet - Ensino-Aprendizagem de Estatística na Web - Mozilla Firefox". The URL is "http://www.sestatnet.ufsc.br/labstat/DescrUni.php". The page title is "Univariado" and the sub-section is "Descrição Univariada". The variable selected is "aluno". The interface includes a sidebar with navigation options: TESTES DE ADEQUÊNCIA, DESCRIÇÃO DE DADOS, TESTES DE HIPÓTESES PARAMÉTRICOS, TESTES DE HIPÓTESES NÃO PARAMÉTRICOS, and MODELOS DE REGRESSÃO. The main content area displays descriptive statistics for the variable "aluno".

Estatísticas Descritivas	
aluno	
Estatísticas	aluno
Observações	80
Média	46.50000
Desvio-padrão	23.23790
Mediana	46.50000
Mínimo	1.00000
Máximo	88.00000
Intervalo	79.00000
1o. Quartil	26.50000
3o. Quartil	66.50000

Figura 16 – Módulo avançado

Esse ambiente, também, tem um módulo denominado auto-avaliação.

Nesse módulo são realizadas diversas inferências ao aluno com questões contempladas em diferentes níveis de abstração. Ou seja, é uma base com questões classificadas em nível básico, intermediário e avançado (WILGES et al., 2007). A Figura 17 exibe a interface do módulo auto-avaliação.

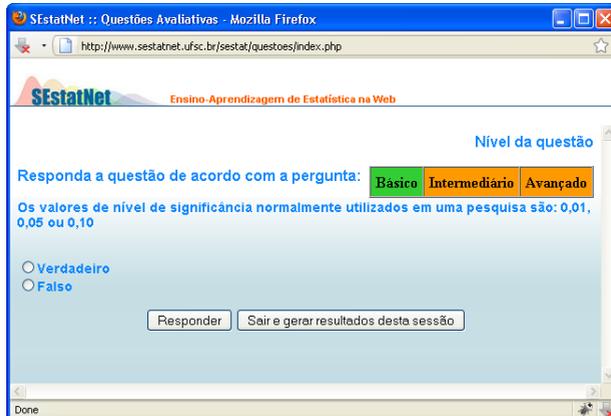


Figura 17 – Módulo auto-avaliação

4.2 EXTENSÃO DO BANCO DE DADOS DO AVEA

O sistema foi implementado com a linguagem de programação PHP (PHP, 2010) e o banco de dados MySQL (MYSQL, 2010). Toda proposta do ambiente SStatNet já foi desenvolvida sobre essas duas tecnologias, incluindo a linguagem de programação *Perl* (PERL, 2010).

Na perspectiva de coletar dados que fossem relevantes para o reconhecimento da aprendizagem do aluno, foram acrescentadas, ao atual banco de dados (BD) do AVEA, duas tabelas que armazenam os registros de *logs*. Esses *logs* representam todo o caminho que os alunos fizeram durante as sessões de aprendizagem. Suas respostas, tempo (segundos) e tomadas de decisões (desempenho prático no AVEA) são registradas nessas tabelas. A representação física dessas tabelas, para o registro das informações dos alunos, tem as características apresentadas na Fig. 18.

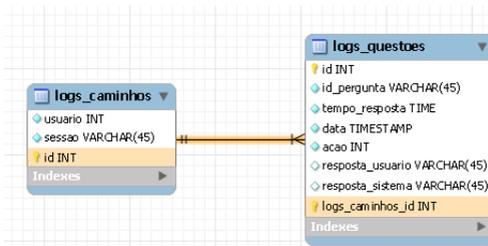


Figura 18 – Modelagem das tabelas para registro dos *logs*

A Modelagem Entidade Relacionamento (MER) destas tabelas foi realizada na ferramenta *Workbench* (WORKBENCH, 2010). Essa ferramenta foi desenvolvida para construção de banco de dados relacionais do MySQL. Na proposta dessa pesquisa foi necessário criar uma tabela que armazenasse o usuário e a sessão em que ele estivesse (*logs_caminho*), também foi criada outra tabela com sua respectiva chave estrangeira de referencia denominada *logs_questoes*. Essa tabela armazena a questão de aprendizagem que o aluno esta analisando, tempo para essa resposta (segundos), a data, a ação (se ele estava avançando ou retrocedendo nas questões), a resposta do aluno e, se for o caso, a resposta do sistema para o desempenho prático.

4.3 AGENTES NO AVEA

A inserção do *framework* proposto dentro desse ambiente, como em qualquer outro AVEA, abrange a identificação dos recursos disponíveis no AVEA. A avaliação teórica no AVEA é composta por n questões em diferentes níveis de aprendizagem. Cada questão tem diferentes números de opções de respostas. A parte prática é caracterizada por uma tomada de decisão do tipo “sim” ou “não” que expressa a interpretação de cálculos realizados no AVEA e que determina a próxima interação do AVEA com o aluno. Os módulos auto-avaliação e tutoria são módulos independentes, desprovidos de uma hierarquia entre as avaliações. Ou seja, o aluno tem total liberdade de realizar as avaliações teóricas e práticas de modo independente e no tempo em que achar conveniente.

Dessa forma existe um mapeamento dos agentes propostos relacionados com os recursos do AVEA. O agente “estudante” é a representação do aluno ativo no AVEA, que contém o motor de inferência

fuzzy para analisar o desempenho do aluno. O agente “avaliação” é identificado por atividades que deem notas ou conceitos ao aluno, como o módulo auto-avaliação. Esse módulo representa uma avaliação teórica do desempenho do aluno. O agente “atividade de aprendizagem” é um agente que atua no processo de ensino-aprendizagem, ele está inserido dentro do módulo de tutoria e extrai os dados desse módulo. Nesse módulo são realizadas perguntas sobre o conteúdo estudado pelo aluno. De modo geral, o módulo de tutoria representa uma avaliação prática do desempenho do aluno. O agente “acompanhamento” é um agente de interface que foi implementado para interagir com o aluno. A Figura 19 apresenta o mapeamento dos agentes do *framework* dentro do AVEA utilizado como estudo de caso.

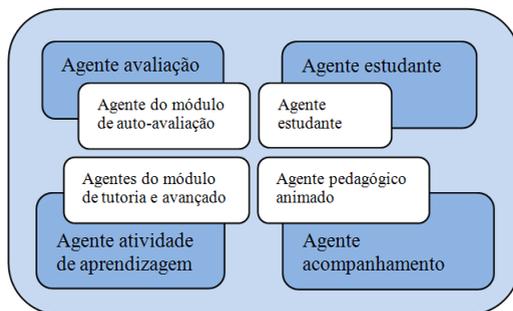


Figura 19 – Mapeamento dos agentes do *framework* para o AVEA

Assim, o protótipo da arquitetura de SMA no AVEA utilizado nessa pesquisa é mostrado na Fig. 20. Inicialmente, os módulos disponíveis no AVEA registram todos os movimentos dos alunos no banco de dados. A seguir, os agentes “avaliação” e “tutoria” filtram os dados e enviam as notas dos alunos, em cada módulo, para a instância do agente “estudante” correspondente. O agente “estudante” analisa as notas de acordo com as crenças disponíveis na biblioteca *fuzzy* e avalia qual desafio pedagógico deve ser ativado para que o agente “acompanhamento” informe o aluno sobre o seu estágio no processo de aprendizagem, de acordo com seu desempenho. A Figura 20 apresenta o fluxo de informação entre os diferentes componentes do sistema.

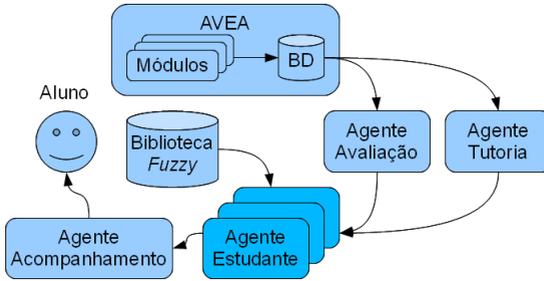


Figura 20 – Protótipo da arquitetura do SMA

4.3.1 A comunicação dos agentes

Alguns dos agentes, implementados nessa pesquisa, são extremamente simples necessitando apenas de consultas a determinadas tabelas do banco de dados do ambiente e envio desses dados ao agente responsável pelo processamento da informação. Nesse contexto inserem-se os agentes reativos: “tutoria” e “avaliação”, pois apenas coletam informações de utilização do ambiente.

A comunicação dos agentes utiliza quatro parâmetros: o agente receptor, o agente emissor, o tipo de ação e o conteúdo da mensagem. As mensagens podem ser destinadas a uma das instâncias do agente “estudante” ou ao agente “acompanhamento”. O tipo de ação define qual mensagem será enviada (ex.: informarNota ou enviarEstimulo). O conteúdo da mensagem carrega dois tipos de informação. A primeira é a nota do aluno nos módulos de tutoria ou auto-avaliação, essa nota pode variar de 0 a 1. A segunda informação é o desafio pedagógico que o agente “acompanhamento” deve executar. A Figura 21 mostra o processo de comunicação entre os agentes.

representadas desse modo, uma vez que as mesmas são desejos com os quais o agente estabeleceu um comprometimento.

Conforme mencionado, os desejos não dirigem necessariamente ao agente agir. Antes de um determinado agente decidir o que fazer, ele confronta os seus desejos com as suas crenças. Dessa forma o agente aplica os desejos, que são possíveis, baseado no desempenho que o aluno atingiu em cada plano.

O processo de deliberação de objetivos é realizado a partir da seleção de um conjunto de desejos que devem ser alcançados, isso de acordo com a situação atual das crenças do agente. Assim, ocorre determinação de como esses desejos concretos ou intenções podem ser atingidos através dos meios disponíveis ao agente. Através dos recursos disponíveis ele emite o desafio pedagógico que foi determinado como um desejo concreto. É nesse momento que é ativado o agente de interface.

4.4 MOTOR DE INFERÊNCIA FUZZY

A implementação de um motor de inferência *fuzzy*, para fins de utilização efetiva, requer o uso de programas que apliquem diretamente as funções da lógica *fuzzy*. Alguns programas de uso geral dispõem de módulos específicos para facilitar a realização desta tarefa, como é o caso do *Fuzzy Logic Toolbox* do MATLAB que contém uma biblioteca disponibilizada através da linguagem C. O MATLAB é um *software* interativo de alta performance voltado para o cálculo numérico. O *toolbox* é um pacote de *software* com uma interface gráfica bem intuitiva para construção e utilização da lógica *fuzzy*.

Neste trabalho optou-se por utilizar o motor de inferência *fuzzy* disponibilizado em código C *stand-alone* do MATLAB. Como o AVEA utilizado nesta pesquisa, e de uma maneira geral quase todos os AVEA são implementados em PHP, considerou-se conveniente e interessante a integração do PHP com as bibliotecas em C do MATLAB. A linguagem C tem a vantagem de oferecer portabilidade entre máquinas e sistemas operacionais, além de um código compacto e rápido (SCHILDT, 1997).

Nesse *toolbox*, disponibilizado pelo MATLAB, existem dois arquivos em C denominados *fismain.c* e *fis.c*, que são fornecidos como códigos-fonte para o motor de inferência *fuzzy* no modo *stand-alone*. O código *stand-alone* em C para o motor de inferência *fuzzy* pode ler um arquivo FIS, que é a extensão dos arquivos *fuzzy* gerados no MATLAB, e um arquivo que contém os dados de entrada das variáveis *fuzzy*.

4.4.1 Modelagem *fuzzy*

As entradas da modelagem são os dados recebidos do agente “avaliação”, ou seja, os dados denominados como desempenho teórico e os dados recebidos pelo agente “tutoria”, denominados como desempenho prático.

As duas variáveis de entrada (desempenhoPratico e desempenhoTeorico) e a variável de saída (avaliacaoGeral) do sistema *fuzzy* são associadas a um conjunto de funções de pertinência. Essas funções são compostas por três variáveis linguísticas: regular, bom e excelente. Foi utilizada a função gaussiana, pois esse tipo de função de pertinência tem um decaimento suave e não tão abrupto como, por exemplo, nas funções triangulares.

O controlador do motor de inferência adotado foi o de *Mamdani*, implementado com o operador de implicação “mínimo” e método de desfuzzyficação adotado foi o método do centro de gravidade. Foram definidas nove regras de inferência que abrangem todas as variáveis linguísticas utilizadas e cada regra é composta pelo operador “and” associado ao método “mínimo”. A agregação entre as regras é feita pelo método “máximo”. Na Figura 22 é apresentada a base de regras.

1. If (desempenhoPraticoAVA is Regular) and (desempenhoTeoricoAVA is Regular) then (avaliacaoGeral is Regular) (1)
2. If (desempenhoPraticoAVA is Regular) and (desempenhoTeoricoAVA is Bom) then (avaliacaoGeral is Regular) (1)
3. If (desempenhoPraticoAVA is Regular) and (desempenhoTeoricoAVA is Excelente) then (avaliacaoGeral is Bom) (1)
4. If (desempenhoPraticoAVA is Bom) and (desempenhoTeoricoAVA is Regular) then (avaliacaoGeral is Regular) (1)
5. If (desempenhoPraticoAVA is Bom) and (desempenhoTeoricoAVA is Bom) then (avaliacaoGeral is Excelente) (1)
6. If (desempenhoPraticoAVA is Bom) and (desempenhoTeoricoAVA is Excelente) then (avaliacaoGeral is Excelente) (1)
7. If (desempenhoPraticoAVA is Excelente) and (desempenhoTeoricoAVA is Regular) then (avaliacaoGeral is Bom) (1)
8. If (desempenhoPraticoAVA is Excelente) and (desempenhoTeoricoAVA is Bom) then (avaliacaoGeral is Excelente) (1)
9. If (desempenhoPraticoAVA is Excelente) and (desempenhoTeoricoAVA is Excelente) then (avaliacaoGeral is Excelente) (1)

Figura 22 – Regras de inferência

Para ativar o motor de inferência, o agente “estudante” gera um arquivo com o conjunto de entradas que contém o desempenho teórico e o desempenho prático dos alunos. Após a geração desse arquivo, o mesmo é submetido juntamente com o modelo *fuzzy* (arquivo FIS) ao motor de inferência, que retorna a variável de saída (avaliacaoGeral) desfuzzyficada.

4.4.1.1 Biblioteca de níveis

A modelagem *fuzzy* contempla uma biblioteca com quatro níveis. Os níveis foram definidos com diferentes valores para os parâmetros das funções de pertinência. Dessa forma, foi possível desenvolver uma base de crenças que permitisse a aplicação de um conjunto de funções de pertinência que variam dinamicamente conforme adaptação e desempenho do aluno. Assim, foram implementadas funções de pertinência que alternam de uma maior incerteza a uma menor incerteza. O objetivo foi construir uma função inicial mais flexível, ou seja, a categorização de um aluno dentro dos conjuntos regular, bom e excelente não é tão restritiva. A Figura 23 apresenta o primeiro nível da biblioteca.

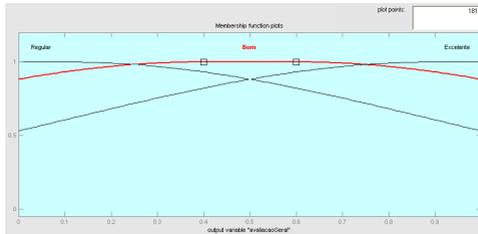


Figura 23 – Nível *fuzzy* 1: Maior incerteza

A proposta desse nível foi caracterizar a maior incerteza presente no início do processo de ensino-aprendizagem do AVEA. As funções de pertinência do primeiro nível da biblioteca apresentam-se ampliadas no intuito de representar um modelo mais adequado ao reconhecimento inicial do aluno dentro do processo de ensino-aprendizagem. Com o objetivo de verificar se o aluno está pronto para avançar para outro nível, o agente “estudante” avalia o valor da saída defuzzificada de cada nível. Assim, o aluno é considerado apto para mudar de nível se estiver acima de 60% do valor máximo possível nesse nível, então o sistema atualiza a base de crenças para o próximo nível. Esse valor mínimo da avaliação geral é um parâmetro institucional e foi adotado como 60%, pois nesta instituição, a nota mínima para aprovação é 6.0. Nas Figuras 24 e 25 são apresentados os níveis 2 e 3. Esses níveis representam, respectivamente, graus de incerteza moderada e pequena. Assim, o agente pode definir, com maior precisão, o conhecimento do aluno a respeito do conteúdo estudado.

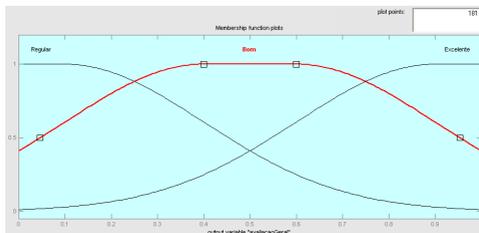


Figura 24 – Nível *fuzzy* 2: Incerteza moderada

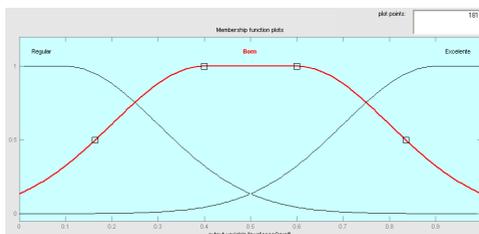


Figura 25 – Nível *fuzzy* 3: Incerteza pequena

A Figura 26 apresenta o último nível proposto nessa biblioteca. O nível 4 é a fase final do processo, é o nível que tem a menor incerteza na avaliação do desempenho do aluno. Esses níveis formam a base de crenças que o agente “estudante” tem sobre os alunos no AVEA. Assim, os níveis podem ser alterados dinamicamente, e os parâmetros das funções de pertinência são modificados para caracterizar desde a maior incerteza até a menor incerteza sobre o desempenho do aluno.

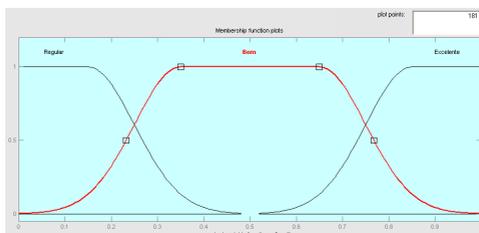


Figura 26 – Nível *fuzzy* 4: Menor incerteza

4.5 AGENTE DE INTERFACE

Todo o processo de comunicação com o aluno é realizado pelo agente “acompanhamento”, que é um agente de interface. O agente “acompanhamento” recebe o tipo de desafio, que deve ser enviado ao aluno, do agente “estudante” após o processo de inferência. Esse agente é representado por um personagem animado e atua sempre que existe um desafio disponível durante a mudança de níveis da biblioteca *fuzzy*. A Figura 27 mostra a representação gráfica do agente “acompanhamento” indicando o aluno a utilizar o módulo avançado do AVEA. Esse exemplo de desafio pedagógico ocorre depois que o aluno passa por todos os níveis disponíveis na biblioteca *fuzzy*.

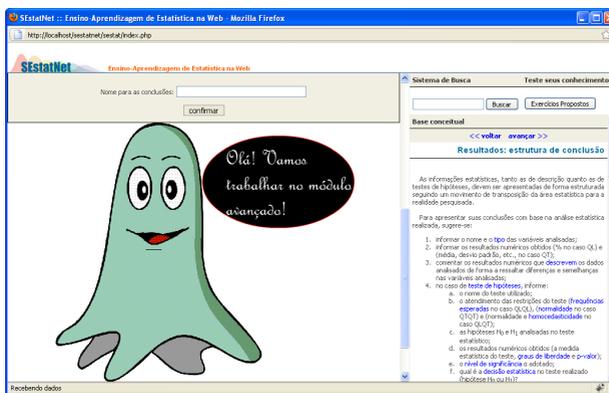


Figura 27 – Visualização do agente de interface

O agente de interface foi implementado para dar características motivacionais aos módulos do AVEA. Seu desenho é inspirado na distribuição bivariada, isto é, constitui a distribuição conjunta de duas distribuições univariadas Gaussianas (WILGES, 2008).

4.6 RESULTADOS

Nessa seção são apresentados alguns dos resultados obtidos de uma amostra de 87 alunos que utilizaram o AVEA entre o primeiro semestre de 2010 até a metade do segundo semestre de 2010. Os alunos observados são estudantes de Engenharia e Automação da Universi-

dade Federal de Santa Catarina, que estavam cursando a disciplina de Estatística.

A Figura 28 apresenta uma amostra da tabela na base de dados do AVEA com os desempenhos dos alunos. Na tabela são armazenados todos os níveis em que o aluno já esteve, sua nota da avaliação geral produzida pelo FIS, os desempenhos máximo e mínimo para o nível, além da identificação do aluno e a data de realização da avaliação.

The screenshot shows a MySQL database window with the following table structure and data:

id	idu	id_bib	nota	max	min	data
275	3241	1	0,485709	0,55021	0,485709	2010-11-01 14:02:53
276	3241	1	0,485709	0,55021	0,485709	2010-11-01 14:03:51
280	3241	1	0,485709	0,55021	0,485709	2010-11-01 16:39:59
282	3241	1	0,485709	0,55021	0,485709	2010-11-01 17:06:33
283	3241	1	0,485709	0,55021	0,485709	2010-11-01 17:14:52
284	3241	1	0,485709	0,55021	0,485709	2010-11-01 17:17:33
198	3243	1	0,529232	0,55021	0,485709	2010-10-16 22:26:07
224	3243	2	0,692632	0,725968	0,590081	2010-10-29 09:05:56
226	3243	3	0,790784	0,803028	0,278549	2010-10-29 10:37:35
228	3243	4	0,85663	0,857164	0,145149	2010-10-29 10:43:29
229	3243	4	0,85663	0,857164	0,145149	2010-10-29 10:56:07
230	3243	4	0,85663	0,857164	0,145149	2010-10-29 11:00:19
231	3243	4	0,85663	0,857164	0,145149	2010-10-29 11:13:20
232	3243	4	0,85663	0,857164	0,145149	2010-10-29 11:17:25
233	3243	4	0,85663	0,857164	0,145149	2010-10-29 11:31:46
234	3243	4	0,852883	0,857164	0,145149	2010-10-29 11:29:07
235	3243	4	0,85663	0,857164	0,145149	2010-10-29 11:30:00

Figura 28 – Tabela com o desempenho dos alunos

Foram gerados gráficos para fazer a análise geral e específica do comportamento do *framework* em relação a sua utilização pelos alunos. Nesses gráficos são mostradas as 10 primeiras interações do aluno com o AVEA, isto é, sempre que o aluno chega ao final de uma análise estatística. As análises são realizadas no módulo de tutoria e o agente “estudante” avalia os resultados no final do processo.

O gráfico do acompanhamento dos alunos é mostrado na Fig. 29. Nesse gráfico foram acompanhados cinco alunos, cada linha representa o andamento de um aluno no ambiente. É possível observar que alguns alunos atingem o último nível nas primeiras interações, pois eles utilizam os recursos disponíveis no AVEA e seu desempenho é muito bom. Também pode-se perceber que existem alunos que demoram mais para atingir outros níveis, mas acabam melhorando o desempenho e alcançam os níveis mais elevados. Os alunos que não saem dos primeiros níveis não atingiram o desempenho esperado, pois não usaram os recursos do ambiente ou obtiveram notas muito baixas.

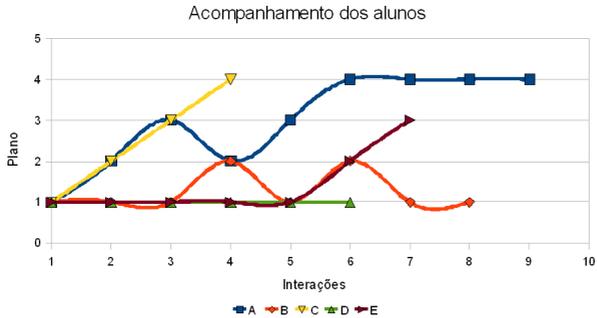


Figura 29 – Acompanhamento dos alunos

Foi realizada uma análise comparativa do acompanhamento do aluno no ambiente e no ensino presencial. Nessa análise foram relacionadas as avaliações feitas em sala de aula com o histórico de desempenho dos alunos. São apresentados cinco casos de alunos com desempenhos diferentes, variando de regular a excelente. Nos alunos com desempenho regular foi constatada a necessidade de implementar desafios para os níveis iniciais, visto que esses alunos não avançam no processo de aprendizagem e não possuem notas boas. Em outros casos, nos alunos com melhor desempenho observa-se que a nota presencial é mais elevada. A Figura 30 mostra alguns exemplos de alunos que utilizaram o AVEA e o seu desempenho foi avaliado pelo sistema.

Relatório de acompanhamento dos alunos													
Aluno	Deslocamento nos planos								Observações	Notas presenciais			
3249	1	1	1	2	3					Progresso moderado	7,3	9,6	8,5
3243	1	2	3	4	4	4	4	4	4	Desempenho excelente	8,3	9,0	9,2
3223	1	1	1							Necessidade de estímulo nos planos iniciais	5,0	4,5	6,0
3331	1	2	3	4	4					Desempenho excelente	10	9,3	10
3240	1	1	1	1	1					Necessidade de estímulo nos planos iniciais	8,4	7,3	5,5

Figura 30 – Análise do acompanhamento dos alunos

A Figura 31 apresenta um gráfico com a distribuição dos alunos nos níveis após 10 interações com o AVEA. Na primeira interação todos os alunos se encontram no nível 1. Na segunda interação alguns alunos já avançam para o nível 2, na terceira e quarta interações os alunos já se distribuem por todos os níveis. Percebe-se que a utilização vai diminuindo a medida que o número de interações aumenta, isso é um processo natural pois os alunos já realizaram suas pesquisas e deixam de utilizar o ambiente.

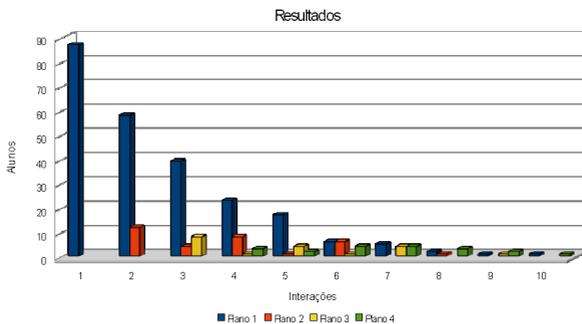


Figura 31 – Distribuição dos alunos nos níveis

Na continuação desta pesquisa é possível criar mais desafios para que o agente de acompanhamento realize outras interações com os alunos. Assim, os alunos recebem desafios mais pontuais para participar de todos os recursos disponíveis no AVEA, melhorando o processo de aprendizagem. Além disso, é possível que sejam criados outros agentes que colaborem com o acompanhamento do desempenho do aluno, para diminuir a incerteza presente na definição do modelo de aluno.

Os resultados dessa implementação, que foram obtidos no primeiro e segundo semestres de 2010, mostram que um sistema de acompanhamento do desempenho é interessante e pode estimular o processo de ensino-aprendizagem dos alunos.

5 CONCLUSÕES

Muitos AVEA têm a preocupação de construir uma arquitetura de agentes que possuam a capacidade de raciocínio. Assim, a proposta da arquitetura BDI, que modela agentes com crenças, desejos e intenções, viabiliza a implementação destes agentes em um AVEA que é diretamente utilizado pelos alunos. Os agentes BDI permitem gerenciar os recursos do AVEA, realizando um movimento de aprendizagem que é direcionado para diferentes módulos do AVEA de acordo com o desempenho do aluno.

O agente “estudante” percebe as dificuldades do aluno por meio do reconhecimento de seu desempenho nas diferentes atividades propostas no AVEA. O uso de LF na análise do desempenho pode flexibilizar a categorização do processo de ensino-aprendizagem do aluno. Não sendo, simplesmente, um tratamento “*crisp*” do tipo certo e errado e sim um valor lógico difuso que pode variar no intervalo entre 0 e 1. Depois desse processo de fuzzyficação, o agente “estudante” comunica o agente “acompanhamento” para informar o recurso mais adequado a ser ativado. Este processo ocorre de forma dinâmica e transparente a cada ação do usuário no ambiente.

A integração da LF com a arquitetura de agentes BDI possibilita que os agentes tenham a capacidade de lidar com informações incertas em um modelo construído a partir de estados mentais. Essa integração permite que a incerteza presente no processo de análise do desempenho dos alunos seja considerada pelos agentes envolvidos. Além disso, os resultados dessas análises foram mais adequados ao perfil de desempenho de cada aluno analisado.

Em trabalhos relacionados verificou-se uma integração envolvendo a arquitetura de agentes BDI com redes Bayesianas (FAGUNDES, 2007). Nesse trabalho os estados mentais foram representados por um modelo probabilístico que trata a incerteza da informação. De maneira similar a integração de tecnologias proposta por Fagundes (2007), essa pesquisa optou pelo uso da LF que permitiu a flexibilização da análise do desempenho dos alunos, pois a definição das funções de pertinência torna o processo de classificação menos rígido, já que não utiliza um conjunto de probabilidades a priori.

A análise do *framework* implementado mostra que é possível garantir o acompanhamento adequado das atividades teóricas e práticas. Além disso, a implementação de outros desafios pedagógicos pode contribuir com a aprendizagem no AVEA. Isso acontece porque o *fra-*

mework proposto resgata, dentro do AVEA, características essenciais do processo de aprendizagem. Como, por exemplo, prender a atenção do aluno, avaliar o aluno para verificar se ele entendeu os conceitos apresentados, além de enviar mensagens ao aluno sugerindo que ele avance dentro do AVEA, pois o mesmo possui um desempenho favorável.

Como trabalhos futuros é possível integrar as mesmas técnicas descritas nesta pesquisa em outros AVEA como, por exemplo, o *Moodle* (MOODLE, 2010). Já que muitas instituições de ensino e centros de formação estão adaptando a plataforma *Moodle* aos próprios conteúdos, com sucesso, não apenas para cursos totalmente à distância, mas também como apoio aos cursos presenciais (WIKIPEDIA, 2010).

REFERÊNCIAS

- ADILEA, W. *Extração de Conhecimento a partir de Redes Neurais aplicada ao Problema da Cinemática Inversa na Robótica*. Dissertação (Mestrado) — USP, 2003.
- ALVARES, L.; SICHMAN, J. Introdução aos sistemas multiagentes. *Jornada de atualização em Informática, 16.; Congresso da SBC, 17.*, Brasília, p. 1–38, 1997.
- BARDIN, L. *Análise de conteúdo*. Trad. luís antero reto e augusto pinheiro. [S.l.: s.n.], 2002.
- BAVARESCO, N.; SILVA, J. M. C. da; SILVEIRA, R. A. Modelo de agentes bdi para criação de objetos inteligentes de aprendizagem. *Simpósio Brasileiro de informática na Educação - SBIE*, 2008.
- BATMAN, M. E. *Intention, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press, 1987.
- BRENNER, W.; RÜDIGER, Z.; WITTIG, H. *Intelligent Software Agents: Foundations and applications*. Berlin: Springer-Verlag, 1998.
- BROOKS, R. A. A robust layered control system for a mobile robot. *In IEEE Journal of Robotics and Automation*, p. 14–23, 1986.
- COSTA, J. C. *Aplicação da Lógica Fuzzy na Construção de um Modelo de Cadeia Ising*. [S.l.]: Universidade Federal de Lavras, 2006.
- COSTA, M. T. C. *Uma Arquitetura Baseada em Agentes para Suporte ao Ensino à Distância*. Tese (Doutorado) — UFSC, 1999.
- FABRI, J.; FABRI, M. Ferramenta fuzzy para acompanhamento do desempenho dos alunos nos cursos à distância. *Anais do XXII Congresso da SBC - WIE*, 2002.
- FAGUNDES, M. Um ambiente para desenvolvimento de agentes bdi. *Trabalho de Conclusão de Curso*, 2004.
- FAGUNDES, M. *Integrating BDI Model and Bayesian Networks*. Dissertação (Mestrado) — UFRGS, 2007.
- FERBER, J.; GASSER, L. Intelligence artificielle distribuée. *International Workshop on Expert Systems and their Applications*, 1991.

GEORGEFF, M. et al. The belief desire intention model of agency. In: *In Proceedings of the 5th International Workshop on Intelligent Agents V: Agent Theories, Architectures, and Languages*. Heidelberg, Germany: Springer-Verlag, 1999.

GIRAFFA, L. *Uma arquitetura de tutor utilizando estados mentais*. Tese (Doutorado) — Ciência da Computação - CPGCC/UFRGS, Porto Alegre, 1999.

GUEDES, V. S. *Formação de Grupos em Ambientes de Ensino Colaborativo Apoiados por Computador*. [S.l.]: Universidade Federal de Pernambuco, 2005.

HADJILEONTIADOU, S. et al. On enhancing on-line collaboration using fuzzy logic modeling. In: *Educational Technology & Society*. [S.l.: s.n.], 2004. p. 68–81.

HUANG, C.-J. et al. An intelligent learning diagnosis system for web-based thematic learning platform. In: *Computers & Education*. [S.l.: s.n.], 2007.

JAMESON, A. Numerical uncertainty management in user and student modeling: An overview of systems and issues. *User Modeling And User-Adapted Interaction*, p. 193–251, 1995.

JENNINGS, N. R.; SYCARA, K.; WOOLDRIDGE, M. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, v. 1, p. 275–306, 1998.

KLIR, G. J.; CLAIR, U. H. S.; YUAN, B. *Fuzzy Set Theory – Foundations and Applications*. [S.l.: s.n.], 1997.

KLIR, G. J.; YUAN, B. *Fuzzy Sets and Fuzzy Logic Theory and Applications*. [S.l.: s.n.], 1995.

MAMDANI, E. H. Applications of fuzzy algorithms for control of simple dynamic plant. *IEEE*, v. 12, p. 1585–1588, 1973.

MATLAB. *Matrix Laboratory*. Acesso: ago. 2010. <<http://www.mathworks.com/>>.

MENDEL, J. M. *Uncertain rule-based fuzzy logic systems: introduction and new directions*. Upper Saddle River, NJ: [s.n.], 2001.

MOODLE. *Moodle*. Acesso: ago. 2010. <<http://moodle.org/>>.

MOULIN, B.; CHAIB-DRAA, B. *Foundations of distributed artificial intelligence. Cap. 1 - An Overview of Distributed Artificial Intelligence*. [S.l.]: John Wiley and Sons, 1996.

MUKAIDONO, M. *Fuzzy Logic for beginners*. [S.l.]: World Scientific Publishing Company, 2001.

MYSQL. *MySQL*. Acesso: ago. 2010. <<http://mysql.com/>>.

NOLAN, J. An expert fuzzy classification system for supporting the grading of student writing samples. In: *Expert Systems with Applications*. [S.l.: s.n.], 1998. p. 59–68.

OLIVEIRA, C. A. et al. The sestatnet perspective - from a statistical applied tool towards a whole educational tool. In: *9th International Conference on Engineering Education*. [S.l.]: ICEE, 2006.

OLIVEIRA, F. M.; VICARI, R. M.; CERON, R. F. Are learning systems distributed or social systems. *European Conference on AI in Education*, Lisbon, 1996.

PADGHAM, L.; WINIKOFF, M. Prometheus: A pragmatic methodology for engineering intelligent agents. In: *Proceedings of the OOPSLA 2002 Workshop on Agent-Oriented Methodologies*. Seattle: [s.n.], 2002. p. 97–108.

PEDRYCZ, W. Computational intelligence in web-based education: A tutorial. *Journal of Interactive Learning Research*, 2004.

PEDRYCZ, W.; GOMIDE, F. *An Introduction to Fuzzy Sets Analysis and Design*. [S.l.]: The MIT Press, 1998.

PERL. *Practical Extraction and Report Language*. Acesso: ago. 2010. <<http://www.perl.org/>>.

PHP. *Hypertext Preprocessor*. Acesso: ago. 2010. <<http://php.net/>>.

PIVA, J. D. et al. Auxiliar: Um sistema inteligente para cursos online. *Revista Brasileira de Informática na Educação - RBIE*, v. 13, 2005.

PROMETHEUS. *Prometheus Design Tools (PDT)*. Acesso: set. 2010. <<http://www.cs.rmit.edu.au/agents/pdt/>>.

RABELLO, R. S.; REATEGUI, E.; COSTI, J. C. G. Um agente de interface aplicado ao ensino de equações de primeiro grau. *Revista de Novas Tecnologias na Educação*, v. 5, 2007.

RAO, A. S.; GEORGEFF, M. Bdi agents: from theory to practice. In: *In Proceedings of the First International Conference on Multi-Agent Systems*. San Francisco, CA: [s.n.], 1995. p. 312–319.

RASMANI, K. *A Data-Driven Fuzzy Rule-Based Approach for Student Academic Performance Evaluation*. Tese (Doutorado) — Central for Intelligent Systems and their Applications. School of Informatics - Universidade de Edinburgo, Edingurgo, U.K, 2002.

REATEGUI, E. et al. Um agente animado sócio-afetivo para ambientes de aprendizagem. *Revista Brasileira de Informática na Educação – Edição Especial Aprendizagem Colaborativa v. 14 n.3.*, p. 27–38, 2006.

RIBEIRO, A. *A avaliação da aprendizagem: aplicação de um modelo fuzzy para se obter notas mais justas na disciplina de língua portuguesa*. Dissertação (Mestrado) — Programa de Pós-Graduação em Educação Brasileira, Universidade Federal do Ceará, Fortaleza, 2007.

RIEDER, R.; BRANCHER, J. Aplicação da lógica fuzzy a jogos didáticos de computador – a experiência do mercado gl. *VII Congresso Iberoamericano de Informática Educativa*, 2004.

RISSOLI, V.; GIRAFFA, L. Aprendizagem significativa apoiada por um sistema tutor inteligente usando lógica fuzzy. *Workshop do SBIE*, 2006.

RONCARELLI, D.; CATAPAN, A. H. On the wings of icarus: Rheomodus of pedagogical work: building a taxonomy for the choice of a virtual teaching-learning environment vtle. In: 9TH IFIP WORLD CONFERENCE ON COMPUTERS IN EDUCATION (WCCE). *Education and Technology for a Better World*. Bento Gonçalves, RS, Brazil, 2009.

RUSSELL, S.; NORVIG, P. *Inteligência Artificial*. [S.l.]: Editora Campus, 2004.

SANTOS, G. J. C. *Lógica Fuzzy – Monografia de conclusão*. [S.l.]: Universidade Estadual de Santa Cruz - UESC, 2003.

SCHILD, H. *C Completo e Total*. 3ed.. ed. [S.l.]: Makron Books, 1997.

SHAW, I. S.; SIMÕES, M. G. *Controle e Modelagem Fuzzy*. 1. ed. São Paulo: Edgard Blücher Ltda, 1999.

SICHMAN, J. *Du raisonnement social chez les agents: une approche fondée sur la théorie de la dépendance*. Tese (Doutorado) — INPG, Grenoble, France, 1995.

SILVA, H. et al. Um sistema baseado na lógica difusa para decidir os conceitos finais dos estudantes críticos. *WEI*, 2008.

SILVEIRA, R. A.; GOMES, E. R.; VICARI, R. Improving interoperability among learning objects using fipa agent communication framework. In: PROFESSIONAL PRACTICE IN ARTIFICIAL INTELLIGENCE. *IFIP World Computer Conference - WCC*. Santiago: Springer, 2006. p. 51–60.

SOUZA, E. P. d. Avaliação formativa em educação a distância via web. *13o Congresso Internacional de Educação à Distância*, 2007.

STATHACOPOULOU, R. et al. Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis. In: *Information Sciences*. [S.l.: s.n.], 2005. p. 237–307.

TAKAGI, T.; SUGENO, M. Derivation of fuzzy control rules from human operator's control action. *IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis*, p. 55–60, 1983.

VAHLDICK, A.; SANTIAGO, R. d.; RAABE, A. L. O uso de técnicas fuzzy em ambientes inteligentes de aprendizagem. *XIX Simpósio Brasileiro de Informática na Educação*, 2008.

VICARI, R. Artificial intelligence and educational systems. *Journal of School n. 2 of Olivais, vol. 1*, 1988.

WEISS, G. *Multiagent systems - a modern approach to distributed artificial intelligent*. Cambridge, MA: [s.n.], 1999.

WEON, S.; KIM, J. Learning achievement evaluation strategy using fuzzy membership function. *31st ASEE/IEEE Frontiers in Education Conference*, October 2001.

WIKIPEDIA. *Wikipedia*. Acesso: ago. 2010.
<<http://www.wikipedia.org/>>.

WILGES, B. *Um Sistema de Acompanhamento dos Estudantes em um Ambiente Virtual de Ensino-Aprendizagem*. Dissertação (Mestrado) — UFSC, 2008.

WILGES, B. et al. Comportamento adaptativo baseado no caminho de aprendizagem do estudante em um ambiente virtual de ensino aprendizagem - avea. In: *Tecnologias na Educação*. Porto Alegre - RS: [s.n.], 2007.

WOOLDRIDGE, M. *Reasoning about Rational Agents*. London: MIT Press, 2000.

WOOLDRIDGE, M. *An Introduction to MultiAgent Systems*. Chichester, UK: Wiley, 2009.

WORKBENCH. *Workbench*. Acesso: ago. 2010.
<<http://wb.mysql.com/>>.

ZADEH, L. A.; KACPRZYK, J. *Fuzzy Logic for the Management of Uncertainty*. [S.l.: s.n.], 1992.

ZAIANE, O.; LUO, J. Evaluating learners' behaviour in a web-based distance learning environment. *Proceedings of ICALT*, 2001.

APÊNDICE A – Artigos Publicados

Mateus, Gustavo P. ; WILGES, Beatriz ; NASSAR, S. M. ; SIVEIRA, R. A.; BASTOS, R. . Reconhecendo as características dos estudantes por meio de um SMA e Lógica *Fuzzy* In: RENOTE - Revista Novas Tecnologias na Educação. ISSN 1679-1916. 2010.

Mateus, Gustavo P. ; WILGES, Beatriz ; NASSAR, S. M. ; SIVEIRA, Ricardo Azambuja . Uma ferramenta de análise do desempenho de estudantes baseada em SMA e Lógica *Fuzzy*. In: Simpósio Brasileiro de Informática na Educação (SBIE), 2009, Florianópolis. SBIE 2009.

Mateus, Gustavo P. ; WILGES, Beatriz ; NASSAR, S. M. ; SIVEIRA, Ricardo Azambuja . *A Belief Desire Intention Multi Agent System in a Virtual Learning Environment*. In: *9th IEEE International Conference on Advanced Learning Technologies*, 2009, Riga, Latvia. ICALT, 2009.

Mateus, Gustavo P. ; WILGES, Beatriz ; WRONSKI, V. ; NASSAR, S. M. ; SIVEIRA, Ricardo Azambuja . Uma arquitetura de Sistemas Multiagentes BDI aplicada em um Ambiente Virtual de Ensino-Aprendizagem. Fortaleza. Simpósio Brasileiro de Informática na Educação (SBIE), 2008.