

Uma Arquitetura de Agentes para Suporte à Colaboração na Aprendizagem Baseada em Problemas em Ambientes Virtuais de Aprendizagem

Alexandre Ádames Alves Pontes

Dissertação de Mestrado submetida ao Programa de Pós-graduação em
Ciência da Computação da Universidade do Estado do Rio Grande do
Norte e Universidade Federal Rural do Semi-Árido como parte dos re-
quisitos necessários para obtenção do grau de Mestre em Ciência da
Computação.

Prof. Francisco Milton Mendes Neto, Dr.

Orientador

Prof. Gustavo Augusto Lima de Campos, Dr.

Co-Orientador

Mossoró, Rio Grande do Norte, Brasil, Junho de 2010

Uma Arquitetura de Agentes para Suporte à Colaboração na Aprendizagem Baseada em Problemas em Ambientes Virtuais de Aprendizagem

Alexandre Ádames Alves Pontes

Dissertação de Mestrado apresentada em Junho de 2010

Prof. Francisco Milton Mendes Neto, Dr.

Orientador

Prof. Gustavo Augusto Lima de Campos, Dr.

Co-Orientador

Prof. Francisco Milton Mendes Neto, Dr.

Componente da Banca

Prof. Gustavo Augusto Lima de Campos, Dr.

Componente da Banca

Prof. Jerffeson Teixeira de Souza, Dr.

Componente da Banca

Prof. Pedro Fernandes Ribeiro Neto, Dr.

Componente da Banca

Mossoró, Rio Grande do Norte, Brasil, Junho de 2010

Dedicatória

Dedico esta dissertação ao Senhor Jesus Cristo que sempre está comigo e me ajuda na minha caminhada e à minha mãe Maria Rita Pontes pelo amor e dedicação e ajuda em todos os momentos da minha vida.

Agradecimentos

No princípio era apenas um sonho, agora vejo ele se tornando realidade, muitas foram as lutas, mas também muitas foram as vitórias.

Agradeço ao meu Deus, pois em tudo vi a mão Dele ao meu favor, no início achei que não iria conseguir, mas aos poucos vi Deus abrindo as portas e me dando vitórias, por isso agradeço a Ele pela sua bondade, ajuda e misericórdia sobre a minha vida e por este grande presente, e principalmente por ter colocado pessoas tão especiais na minha vida que me ajudaram na realização desse sonho.

Agradeço a toda minha família que esteve comigo durante este período, mas em especial aos meus pais Maria Rita Pontes e Antônio Alves Pontes, pelo amor, carinho, dedicação e incentivo dados a mim em todos os momentos.

Agradeço ao meu orientador o professor Milton, por ter acreditado em meu potencial, pelo incentivo e principalmente pela paciência de me corrigir até o ponto deste trabalho ficar bom. Ao meu co-orientador professor Gustavo, pelo empenho e ajuda que foram essenciais para o cumprimento deste trabalho.

Gostaria de agradecer aos meus colegas de mestrado que estiveram comigo nesta caminhada e que passaram pelos mesmos apertos. Aos meus colegas do LES que estiveram nesta jornada e em muito contribuíram para o cumprimento deste trabalho, aos meus amigos Ferdinand, Luis e especialmente a minha querida amiga Laysa, que trabalhou mais de perto comigo e muitas vezes esteve ao meu lado em momentos difíceis da caminhada e em muito contribui para o sucesso deste trabalho.

Aos meus colegas da SUTIC, que estiveram junto comigo nesta jornada, pela compreensão por algumas ausências, a eles meus sinceros agradecimentos.

Aos professores do MCC que tanto se empenharam nas disciplinas, para que tivéssemos uma boa formação.

Aos coordenadores do MCC, inicialmente prof. Milton e prof. Marcelino e agora prof. Iguatemi e professora Cláudia que tanto se empenharam para construir esse mestrado e tornar nossa estada nele a melhor possível, também agradeço a Rosita e Marilha, secretárias do MCC que se dedicaram e nos auxiliaram em todo esse processo. Enfim a todos que direta ou indiretamente contribuíram para o sucesso deste trabalho, agradeço a todos de coração.

Resumo

A aprendizagem colaborativa no Ensino a Distância (EaD) é uma modalidade de ensino e aprendizagem que tem crescido e apresentado bons resultados. Um dos grandes desafios atuais é o suporte informatizado a esta atividade. Um ambiente virtual de aprendizagem colaborativa apoiada por computador pode possibilitar aos alunos de cursos a distância baseados na Web interagirem entre si e com um ou mais facilitadores para a realização de trabalhos em grupo. A aprendizagem baseada em problemas (*Problem-Based Learning - PBL*) é uma teoria de aprendizagem que enfatiza a colaboração e o trabalho em grupo para resolução de um problema e tem apresentado resultados significativos no que tange a este novo paradigma de aprendizagem. Contudo a implantação da teoria de aprendizagem baseada em problemas não é uma tarefa trivial. Em ambientes de aprendizagem colaborativa para EaD, a complexidade de implantação desta teoria de aprendizagem é ainda maior, pois o facilitador nem sempre pode detectar possíveis problemas na colaboração, nem possui todas as informações necessárias para aplicar as técnicas de aprendizagem desta teoria. Desta forma, este trabalho apresenta uma arquitetura de agentes para suporte à colaboração na aprendizagem baseada em problemas em ambientes virtuais de aprendizagem, com o objetivo de detectar e corrigir problemas inerentes à implantação desta teoria de aprendizagem, melhorando o processo de aprendizagem.

Palavras-chave: Aprendizagem Colaborativa, Aprendizagem Baseada em Problemas, Agentes Inteligentes, Ambientes Virtuais de Aprendizagem

Abstract

Collaborative learning in Distance Learning (DL) is a mode of teaching and learning that has grown and shown good results. Nowadays one of the main challenges is the computerized support to this activity. A computer-supported collaborative virtual learning environment may enable the students of web-based distance education courses interact among themselves and with one or more facilitators to carry out group work. The Problem-Based Learning (PBL) is a learning theory that emphasizes collaboration and teamwork to solve a problem and has shown significant results in regard to this new learning paradigm. However the implementation of the problem-based learning theory is not a trivial task. In collaborative learning environments for distance education, the complexity of implementation of this learning theory is even higher, because the facilitator can not always detect possible problems in cooperation, nor he has all the information needed to apply the learning techniques of this theory. Thus, this work presents an agent-based architecture to support collaboration in problem-based learning in virtual learning environments, in order to detect and correct problems inherent to the implementation of this learning theory, improving the learning process.

keywords: Collaborative Learning, Problem-Based Learning, Intelligent Agents, Virtual Learning Environments

Sumário

1	Introdução	1
1.1	Contextualização	1
1.2	Problemática	2
1.3	Objetivo Geral	4
1.4	Objetivos específicos	4
1.5	Organização da Dissertação	4
2	Agentes e Sistemas Multiagente	6
2.1	Definição de Agente	6
2.1.1	Agentes Reativos Simples	8
2.1.2	Agentes Reativos com Estado Interno	8
2.2	Sistemas Multiagente	9
2.3	Sistemas Multiagente de Apoio à Aprendizagem	9
2.3.1	COLE (<i>Collaborative Online Learning Environment</i>)	9
2.3.2	GRACILE (<i>Japanese Grammar Collaborative Intelligent Learning Environment</i>) .	10
2.3.3	I-Help	11
2.3.4	FLE (<i>Future Learning Environment</i>)	11
2.3.5	I-MINDS (<i>Intelligent Multiagent Infrastructure for Distributed System in Education</i>)	12
2.3.6	ICSS (<i>Intelligent Collaborative Support System</i>)	12
3	Aprendizagem Colaborativa e as Teorias de Aprendizagem	14

3.1	Aprendizagem Colaborativa	14
3.2	Teorias de Aprendizagem	15
3.2.1	Construtivismo Piagetiano	15
3.2.2	Teoria Socio-Cultural de Vygotsky	16
3.2.3	Cognição Distribuída	17
3.2.4	Teoria da Flexibilidade Cognitiva	17
3.2.5	Teoria da Aprendizagem Significativa	18
3.2.6	Aprendizagem Baseada em Problemas - <i>Problem-Based Learning</i>	18
3.2.6.1	Ciclo de Desenvolvimento da PBL	19
3.2.6.2	Metas da PBL	19
3.2.6.3	O Problema na PBL	20
3.2.6.4	O Papel do Facilitador na PBL	20
3.2.7	Comparativo das Teorias de Aprendizagem	21
3.2.8	Por que PBL?	21
3.2.9	Requisitos de Ambientes CSCL para aplicação bem sucedida da PBL	22
3.2.10	Sistemas Multiagentes e as Teorias de Aprendizagem	23
3.2.11	Uso de Ontologias em Ambientes de Aprendizagem	24
4	Metodologias para Modelagem de SMAs: um Estudo Comparativo	25
4.1	Metodologias para Modelagem de Sistemas Multiagente	25
4.2	CommonKADS	26
4.3	MAS-CommonKADS	27
4.4	K-Org	28
4.5	Tropos	29
4.6	PASSI	30
4.7	Prometheus	31
4.8	Gaia	32

4.9	ADELFE	33
4.10	MESSAGE	35
4.11	INGENIAS	36
4.12	MaSE	37
4.13	Estudo Comparativo	38
4.14	Justificativa para escolha da metodologia	40
5	Arquitetura de Agentes para Suporte à Colaboração na PBL em AVAs	41
5.1	Fases da metodologia Prometheus	42
5.2	Especificação de Sistema	42
5.2.1	Especificação de Metas	43
5.2.2	Funcionalidades	47
5.2.3	Desenvolvimento de Cenários	53
5.3	Projeto de Arquitetura	59
5.3.1	Decidindo os Tipos de Agentes	59
5.3.1.1	Agrupando funcionalidades	60
5.3.1.2	Revendo agrupamento	61
5.3.1.3	Descrevendo os agentes	62
5.3.2	Descrevendo Interações entre os Agentes	64
5.3.2.1	Protocolo de Detecção de Problemas	65
5.3.2.2	Protocolo de Formação de Grupos	66
5.3.3	Diagrama da Estrutura Global do Sistema	67
6	Estudo de Caso: Detecção de Estudantes Passivos e Fuga de Contexto na PBL	71
6.1	Descrição do ambiente virtual de aprendizagem utilizado no estudo de caso	71
6.2	Estudo de Caso	73
6.2.1	Agente Detector de Problemas	75
6.2.2	Agente Estudante	79

6.2.3	Agente Animado de Interface	80
6.3	Conclusão	82
7	Considerações finais e Trabalhos Futuros	83
A	Descrição detalhada do ambiente e-Grupo	86
A.1	Ferramentas de Coordenação	86
A.2	Ferramentas de Cooperação	88
A.3	Ferramentas de Comunicação	90
A.4	Implementação do e-Grupo	91
A.5	Cenário de Uso do e-Grupo	92
	Referências bibliográficas	93

Lista de Símbolos e Abreviaturas

Abreviatura	Descrição
<i>ADELFE</i>	Atelier de Développement de Logiciels à Fonctionnalité Emergente
<i>AUML</i>	Agents Unified Modeling Language
<i>COLE</i>	Collaborative Online Learning Environment
<i>CSCL</i>	Computer-Supported Collaborative Learning
<i>EaD</i>	Ensino a Distância
<i>FLE</i>	Future Learning Environment
<i>GRACILE</i>	Japanese Grammar Collaborative Intelligent Learning Environment
<i>I – MINDS</i>	Intelligent Multiagent Infrastructure for Distributed System in Education
<i>ICSS</i>	Intelligent Collaborative Support System
<i>MaSE</i>	Multiagent Systems Engineering
<i>OSACA</i>	Open System for Asynchronous Cognitive Agents
<i>PASSI</i>	Process for Agent Societies Specification and Implementation
<i>PBL</i>	Problem-Based Learning
<i>PBL</i>	Project-Based Learning
<i>PDT</i>	Prometheus Design Tool
<i>PTK</i>	PASSI Tool Kit
<i>RUP</i>	Rational Unified Process
<i>SAAS</i>	System Analysis for Agent Systems
<i>SMA</i>	Sistema Multiagente
<i>UML</i>	Unified Modeling Language

Lista de Tabelas

3.1	Quadro comparativo das teorias de aprendizagem	21
4.1	Parâmetros de comparação das metodologias	39
4.2	Estudo comparativo de metodologias para modelagem de sistemas multiagente	40
5.1	Funcionalidade Definição do Problema	51
5.2	Funcionalidade Análise do Problema	51
5.3	Funcionalidade Gerência do Ciclo	51
5.4	Funcionalidade Gerência de Problemas	52
5.5	Funcionalidade Gerência de Estratégias Pedagógicas	52
5.6	Funcionalidade Gerência de Estudantes	52
5.7	Funcionalidade Formação de Grupos	53
5.8	Funcionalidade Recomendar Pares	53
5.9	Funcionalidade Motivar Estudantes	53
5.10	Cenário Auxílio na Definição do Cenário do Problema	54
5.11	Cenário Criação de Modelos de Estudantes	55
5.12	Cenário Formação de Grupos Afins	55
5.13	Cenário Auxílio na Análise do Problema	56
5.14	Cenário Detecção de Fuga do contexto	57
5.15	Cenário Detecção de Problemas de Colaboração	58
5.16	Cenário Detecção de Estudantes Passivos	58
5.17	Cenário Detecção de Estagnação do Processo	59

5.18	Descritor AgenteFacilitador	62
5.19	Descritor AgenteDetectordeProblemas	63
5.20	Descritor AgenteEstudante	63
5.21	Descritor AgenteGrupo	64
5.22	Descritor AgenteInterface	64
6.1	Tabela de pontuação de participação	75

Lista de Figuras

2.1	Um agente reativo simples	8
2.2	Um agente reativo com estado interno	9
3.1	Ciclo da PBL adaptado de (HMELO-SILVER, 2004)	20
5.1	Fases da metodologia Prometheus adaptado de (PADGHAM; WINIKOFF, 2002)	42
5.2	Diagrama de Metas	46
5.3	Diagrama de Funcionalidades	48
5.4	Legenda do Diagrama de Funcionalidades	49
5.5	Diagrama de Cenários	54
5.6	Diagrama de Acoplamento de Dados	60
5.7	Diagrama de Agente-Funcionalidade	61
5.8	Diagrama de Conhecimento do Agente	61
5.9	Protocolo de detecção de problemas	65
5.10	Protocolo de formação de grupos	66
5.11	Diagrama da estrutura global do sistema	68
6.1	Exemplo da interface web do e-Grupo	72
6.2	Abordagem para detecção de estudantes passivos e fuga do contexto do problema . . .	74
6.3	Parte do código da ontologia em OWL instanciada com algumas palavras relacionadas ao contexto do problema	79
6.4	Parte do código da ontologia em OWL instanciada com algumas palavras fora do contexto do problema	79

6.5	Parte do código fonte do agente estudante em Java	80
6.6	Exemplo do agente animado de interface no e-Grupo	81
6.7	Parte do código fonte do agente animado de interface em Java	81

Capítulo 1

Introdução

1.1 Contextualização

O Ensino a Distância (EaD) é uma modalidade de ensino e aprendizagem que tem crescido e apresentado bons resultados. Um dos grandes desafios atuais é o suporte informatizado a esta atividade. Uma atividade importante existente no ensino presencial é a atividade em grupo. A interação entre os estudantes no desenvolvimento de alguma atividade pedagógica é fundamental para o processo de aprendizagem, pois cada estudante compartilha com os colegas seus conhecimentos, dúvidas e impressões sobre o que foi discutido na aula, enriquecendo assim o processo de aprendizagem, e dando início a uma nova modalidade de aprendizagem, a Aprendizagem Colaborativa (MENDES NETO; BRASILEIRO, 2001) (MENDES NETO; BRASILEIRO, 2002) (COUTINHO; JUNIOR; BATISTA, 2007).

Uma definição bem ampla sobre aprendizagem colaborativa pode ser extraída de (DILLENBOURG, 1999), que define tal processo de aprendizagem como uma situação onde duas ou mais pessoas aprendem ou tentam aprender alguma coisa juntas. A quantidade de pessoas pode ser um pequeno grupo ou uma grande comunidade. O aprendizado pode ser orientado a um curso ou a resolução de um problema.

Na abordagem tradicional, o ensino é centrado no professor. O fundamento de tal postura é a de que o professor detém o conhecimento que deve ser transferido para o estudante. Nesse modo de ensinar, as atividades de aula se desenvolvem sob a única e exclusiva orientação do professor (MARQUES, 2006). Na perspectiva da aprendizagem colaborativa, os estudantes trabalham juntos em pequenos grupos tendo uma única meta (COUTINHO; JUNIOR; BATISTA, 2007). A aprendizagem é realizada de forma coletiva e o estudante é responsável por sua aprendizagem. A discussão de idéias entre os membros do grupo aumenta o interesse e promove o pensamento crítico.

As principais teorias de aprendizagem que apóiam este paradigma de ensino colaborativo se baseiam, principalmente, na teoria sociocultural de Vygotsky (VIGOTSKI, 1998), a qual enfatiza que a inteligên-

cia do indivíduo se origina da sociedade e da cultura, ou seja, de uma interação entre o indivíduo e o ambiente no qual ele está inserido. Esta teoria de aprendizagem enfatiza que o conhecimento humano advém primeiramente do meio, através de relações interpessoais, e não de aspectos internos (intrapessoal) do indivíduo. Além disto, Piaget (PIAGET, 1973) citado por (CASTAÑON, 2005) mostra em seus estudos como o conhecimento é adquirido pelo indivíduo. Segundo ele, o conhecimento não é algo inerente ao indivíduo, mas parte da relação entre o indivíduo e o meio em que ele está inserido. A aprendizagem baseada em problemas (*Problem-Based Learning - PBL*), segundo (HMELO-SILVER, 2004), é um método no qual os estudantes aprendem através da resolução de um problema. A aprendizagem é centrada no estudante e auto-dirigida. Os estudantes trabalham em pequenos grupos colaborativos na resolução de um problema. O professor atua como facilitador do processo de aprendizagem ao invés de apenas passar conhecimentos (HMELO-SILVER; BARROWS, 2006).

A aprendizagem colaborativa destaca a participação ativa e a interação, tanto dos estudantes como dos professores, que passam a ser chamados de facilitadores¹. O conhecimento é visto como uma construção social e, por isso, o processo educativo é favorecido pela participação social em ambientes que propiciem a interação, a colaboração e a avaliação. Pretende-se que os ambientes de aprendizagem colaborativa sejam ricos em possibilidades e propiciem o crescimento do grupo.

O suporte informatizado à aprendizagem colaborativa é chamado de Aprendizagem Colaborativa Apoiada por Computador, do inglês *Computer-Supported Collaborative Learning - CSCL* (DIMITRACOPOULOU, 2005). Nesta modalidade de aprendizagem são usados recursos computacionais para oferecer suporte à aprendizagem colaborativa no ensino a distância, que incluem recursos de comunicação síncrona e assíncrona, promovendo os mesmos benefícios do ensino presencial.

Agentes de software têm sido amplamente utilizados no âmbito educacional, esta tecnologia tem se mostrado bastante promissora como auxílio em ambientes colaborativos de aprendizagem, dinamizando o processo. Eles podem ser usados, por exemplo, para auxiliar no cumprimento de uma dada teoria de aprendizagem em um ambiente colaborativo.

1.2 Problemática

A PBL, quando aplicada de forma correta, oferece benefícios importantes. Dentre os quais destacam-se (HMELO-SILVER, 2004):

- Desenvolve pensamento crítico e criatividade no estudante;

¹Facilitador é o responsável, na modalidade de Ensino a Distância, pelo acompanhamento da evolução da aprendizagem do estudante. É semelhante ao professor no ensino presencial, mas com outro papel, pois enquanto o professor é responsável pela formação do estudante, o Facilitador tem a função de facilitar sua aprendizagem (MENDES NETO, 2000)

- Aumenta sua capacidade de resolução de problemas;
- Aumenta a motivação;
- Ajuda os estudantes a aplicarem os conhecimentos adquiridos em novas situações;
- Torna os estudantes mais ativos, pois eles resolvem problemas do mundo real e são responsáveis por sua própria aprendizagem.

Além disso, Hmelo-Silver e Barrows (2006) mostram estudos empíricos que afirmam que estudantes que aprendem através desta abordagem possuem uma maior capacidade de aplicar seus conhecimentos em problemas novos e usarem de maneira mais eficaz estratégias de aprendizagem de forma auto-dirigida do que estudantes que aprendem através de métodos tradicionais de ensino.

No entanto, a PBL possui algumas dificuldades para ser colocada em prática. Isto se deve ao fato de que esta teoria de aprendizagem possui várias características únicas, e alguns fatores devem existir para o sucesso de sua aplicação. Dentre as principais dificuldades de implementação da PBL, podemos citar:

1. Não é trivial completar o ciclo de desenvolvimento da PBL (HMELO-SILVER, 2004);
2. A construção do cenário do problema não é fácil pois os problemas devem ser complexos e mal estruturados, além de permitirem reflexão por parte dos estudantes (SAVERY, 2006);
3. O processo de aprendizado é complexo pois deve envolver vários objetos de estudos e deve ser multidisciplinar (SAVERY, 2006);
4. É difícil para o facilitador garantir uma colaboração efetiva (SAVERY, 2006);
5. É difícil para o facilitador monitorar o processo de aprendizagem pois este é auto-dirigido pelo estudante, de forma que o facilitador não pode oferecer as respostas, apenas guiá-lo neste processo (SAVERY, 2006);
6. Não é trivial para os estudantes adquirirem as habilidades necessárias após o processo de desenvolvimento da PBL (SAVERY, 2006);
7. O método de avaliação é complexo pois deve avaliar o progresso do estudante seguindo os objetivos específicos da PBL (SAVERY, 2006);
8. É difícil o facilitador possuir as competências específicas para guiar o processo de aprendizagem, como: conhecimentos adequados no que diz respeito ao tema em estudo, vontade de se envolver com os alunos de uma forma autêntica, bem como a habilidade de expressar-se em uma linguagem compreendida pelos estudantes (HMELO-SILVER; BARROWS, 2006);

9. É difícil para o facilitador ter conhecimento profundo das técnicas e estratégias usadas na PBL (HMELO-SILVER; BARROWS, 2006).

A implantação de um método de ensino com base na PBL não é uma tarefa trivial. Em ambientes de aprendizagem colaborativa para EaD, a complexidade de implantação deste método é ainda maior, pois o facilitador nem sempre pode detectar possíveis problemas na colaboração, nem possui todas as informações necessárias para aplicar as técnicas de aprendizagem deste método, como, por exemplo, saber quando os estudantes estão saindo do foco da discussão e tomar medidas de correção (HMELO-SILVER; BARROWS, 2006).

1.3 Objetivo Geral

Tendo em vista a problemática apresentada, este trabalho tem como objetivo propor uma arquitetura baseada em agentes para apoiar a aprendizagem colaborativa, sendo que os agentes serão especificados de forma a atender os requisitos da PBL. Foi escolhido o uso de agentes de software como solução para este trabalho porque a arquitetura visa tornar o ambiente mais pró-ativo, autônomo e dinâmico, permitindo a atuação mais efetiva dos participantes do processo de aprendizagem.

1.4 Objetivos específicos

Como objetivos específicos deste trabalho podemos destacar:

- Avaliação de diversos Sistemas Multiagente de Apoio à Aprendizagem;
- Comparação das principais teorias de aprendizagem;
- Comparação das diversas metodologias de modelagem de sistemas multiagente.

1.5 Organização da Dissertação

Esta dissertação está organizada da seguinte forma: no Capítulo 2 são apresentados aspectos gerais sobre agentes e sistemas multiagentes, bem como experiências de implantação de sistemas de suporte à aprendizagem que usaram agentes. No Capítulo 3 são mostradas definições da aprendizagem colaborativa, apresentando diversas teorias de aprendizagem, mas com uma discussão mais detalhada acerca da PBL. Ao final deste capítulo também são discutidos alguns sistemas que implementaram uma teoria de aprendizagem em particular. O Capítulo 4 apresenta diversas metodologias de modelagem de Sistemas

Multiagente (SMAs), e realiza um estudo comparativo entre elas. Este capítulo tem por objetivo fomentar a escolha de uma metodologia de modelagem que será usada na modelagem dos agentes que comporão a arquitetura proposta. O Capítulo 5 apresenta a Arquitetura de Agentes para Suporte à Colaboração na Aprendizagem Baseada em Problemas em Ambientes Virtuais de Aprendizagem. O Capítulo 6 discute um estudo de caso com a implementação do caso de detecção de estudantes passivos e discussões fora do contexto do problema na aplicação da PBL. E por fim o Capítulo 7 mostra as considerações finais e os trabalhos futuros.

Capítulo 2

Agentes e Sistemas Multiagente

Este capítulo descreve aspectos relacionados a agentes e sistemas multiagente. Na Seção 2.1 é feita uma discussão sobre várias definições adotadas por autores sobre a definição de um agente. A Seção 2.2 apresenta conceitos relacionados a Sistemas Multiagente (SMAs). E por fim, a Seção 2.3 discute vários ambientes de aprendizagem que se utilizaram de agentes de software para atingirem seus objetivos educacionais.

2.1 Definição de Agente

Existem diversas definições para o termo agente, mas não há ainda um consenso sobre o assunto. Podemos entender este conceito a partir de algumas definições de pesquisadores da área. Segundo (HENDERSON-SELLERS; GIORGINI, 2005), agentes são entidades de software ou não que são caracterizadas por serem: autônomas, proativas, situadas e direcionadas a objetivos. Um agente ser autônomo significa que ele pode agir de forma autônoma e definir seu próprio comportamento. Um agente proativo é aquele que não necessita de solicitação externa para agir. Todo agente também está situado em algum ambiente específico. Um agente é orientado a objetivos quando ele tem alguma meta no ambiente ao qual está inserido.

Segundo Russell e Norvig (2003), um agente está situado em determinado ambiente, percebendo-o através dos seus sensores e agindo sobre ele através de seus atuadores. Os autores também introduzem o conceito de agente racional, que é aquele que realiza de forma correta suas tarefas dada uma medida estipulada de performance, que irá avaliar o grau de sucesso dos comportamentos do agente no ambiente em questão.

Russell e Norvig (2003) definem um agente racional como:

Um agente racional é aquele que para cada possível sequência de percepções, ele deve selecionar a ação que maximize sua medida de performance, baseado na sua sequência de percepções e no seu conhecimento do ambiente.

Outra definição de agente, com boa aceitação é a definição de (WOOLDRIDGE, 2002):

Um agente é um sistema computacional que está situado em algum ambiente e que é capaz de efetuar ações autônomas neste ambiente afim de cumprir os objetivos para os quais ele foi feito.

Outra definição bastante aceita está descrita em (MAES, 1995):

Agentes inteligentes são sistemas computacionais residentes em ambientes dinâmicos complexos, os quais percebem e atuam autonomamente neste ambiente e, ao fazê-lo, realizam um conjunto de objetivos e tarefas para os quais foram designados.

Em (RUSSELL; NORVIG, 2003) são definidos os tipos de agentes listados a seguir:

- **Agente reativo:** Agente que escolhe sua ação baseado na percepção atual, ou seja, sem levar em consideração o histórico das percepções. Este é o tipo de agente mais simples.
- **Agente reativo baseado em modelos:** Este tipo de agente mantém internamente um modelo do seu ambiente e, para selecionar uma ação, leva em consideração o histórico das percepções.
- **Agente baseado em objetivos:** Este tipo de agente pondera suas ações sempre levando em consideração a tentativa de alcançar seus objetivos, além de possuir também um conjunto de planos internos que são usados para cumprir estes objetivos.
- **Agente baseados em utilidades:** Este tipo de agente pondera suas ações sempre tentando maximizar uma função de utilidade que avalia seu grau de “felicidade” na escolha de uma determinada ação.

Cada tipo de agente citado acima pode aumentar sua performance com algoritmos de aprendizado, caracterizando o agente baseado em aprendizado (RUSSELL; NORVIG, 2003).

Os agentes ainda podem ser classificados como estacionários ou móveis, persistentes ou não persistentes, reativos ou cognitivos. Uma das classificações mais importantes de agentes são reativos ou cognitivos.

Agentes reativos são agentes simples que percebem mudanças no ambiente e reagem sem qualquer conhecimento de ações prévias. Como estes agentes não têm memória, eles são incapazes de planejar ações futuras.

Agentes cognitivos são mais complexos porque eles têm uma representação explícita tanto do ambiente quanto de outros agentes. Este tipo de agente tem uma memória, que possibilita ele planejar ações futuras baseado em situações que ocorreram anteriormente (LANGLEY; LAIRD; ROGERS, 2009).

As estruturas dos agentes são definidas basicamente de conceitos pro-ativos e reativos. A seguir serão conceitualizados agentes reativos simples e agentes reativos simples com estado interno de acordo com as definições apresentadas por (RUSSELL; NORVIG, 2003) e (WEISS, 1999).

2.1.1 Agentes Reativos Simples

Agentes reativos simples selecionam uma ação baseados na percepção corrente do ambiente, ignorando percepções prévias. Esta estrutura de agentes assume que em qualquer momento: (1) o agente recebe informação, através de sensores, acerca dos estados do ambiente, definido em um conjunto de possíveis estados, $S = \{s_1, \dots, s_n\}$; (2) um subsistema de percepção, $Ver : S \rightarrow P$, mapeia cada estado do ambiente em uma percepção definida em um conjunto de possíveis percepções, $P = \{p_1, \dots, p_m\}$, que são representações de aspectos dos estados de S que são acessíveis para a decisão do agente; (3) um subsistema de tomada de decisão, $Ação : P \rightarrow A$, processa a percepção em P e seleciona uma ação definida em um conjunto de possíveis ações para o agente, $A = \{a_1, \dots, a_k\}$; (4) o agente executa a ação selecionada no ambiente através de atuadores.

A Figura 2.1 mostra a estrutura de um agente reativo simples.

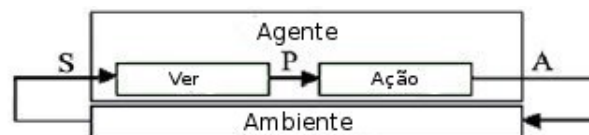


Figura 2.1: Um agente reativo simples

2.1.2 Agentes Reativos com Estado Interno

Para ter um desempenho mais racional, este tipo de agente salva um estado interno com aspectos de controle do mundo que não estão evidentes na percepção corrente. Este estado depende do histórico de percepções prévias do ambiente, sendo definido em um conjunto de possíveis estados internos correntes, $\Delta = \{\delta_1, \dots, \delta_l\}$. A Figura 2.2 mostra a estrutura deste tipo de agente reativo, ilustrando como uma

percepção corrente é combinada com o atual estado interno para gerar uma descrição atualizada do estado corrente.

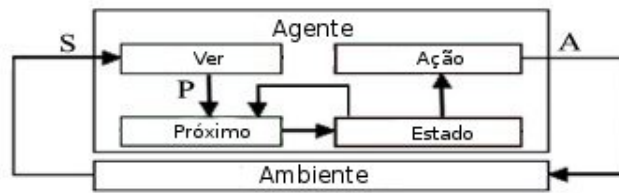


Figura 2.2: Um agente reativo com estado interno

Portanto, há um subsistema, $Próximo : P \times \Delta \rightarrow \Delta$, que mapeia percepções em P e o estado interno corrente em Δ para um novo estado interno em Δ , que será usado para selecionar a próxima ação. A seleção de ações de um agente reativo com estado interno é definida como um mapeamento de estados internos em ações, $Ação : \Delta \rightarrow A$. De acordo com (RUSSELL; NORVIG, 2003), a atualização da informação do estado interno com o passar do tempo requer dois tipos de conhecimento a serem codificados na função *Próximo*: (a) alguma informação acerca de como o mundo evolui independentemente do agente; e (b) alguma informação sobre como as próprias ações do agente afetam o mundo.

2.2 Sistemas Multiagente

Portanto, dado a definição do que seria um agente isolado, podemos introduzir o conceito de Sistema Multiagente (SMA) como aquele composto de agentes cooperativos ou competitivos que interagem entre si para atingir um objetivo individual ou comum (HENDERSON-SELLERS; GIORGINI, 2005). Este conceito de SMA nos leva à percepção de algumas características novas, tais como: comunicação e sociabilidade, ou seja, os agentes necessitam de alguma forma de comunicação para atingir suas metas. Os agentes deverão trabalhar juntos e isso exige um comportamento social dos mesmos.

2.3 Sistemas Multiagente de Apoio à Aprendizagem

Esta seção tem por objetivo discutir diversos trabalhos que utilizaram SMAs em sistemas de apoio à aprendizagem colaborativa.

2.3.1 COLE (*Collaborative Online Learning Environment*)

COLE (AZEVEDO H. ; SCALABRIN, 2005) é um ambiente de apoio à aprendizagem colaborativa que enfoca aspectos sociais da interação entre os participantes do processo de aprendizagem. Tal ambiente

se utiliza da aprendizagem baseada em projetos (*project-based learning* - PBL). Esta abordagem se utiliza de portfólios¹.

A metodologia utilizada pelo autor no desenvolvimento deste ambiente foi a SAAS (*System Analysis for Agent Systems*), descrita em (AZEVEDO, 1997). O ambiente apóia a colaboração entre grupos de no máximo 10 estudantes, cada qual possuindo seu próprio portfólio, que pode esboçar suas idéias e hipóteses e, ao final, submeter à avaliação do grupo a fim de alcançar um consenso.

Esta metodologia possui oito etapas, das quais apenas cinco foram efetivamente utilizadas no desenvolvimento, as quais se seguem: i) descrição dos serviços, onde todos os serviços do sistema foram agrupados em uma tabela e associados ao professor e aos alunos; ii) construção de cenários para cada serviço descrito na primeira etapa; iii) construção de *mock-ups*, que é um recurso usado para mostrar apenas a interface com o usuário; iv) distribuição de competências, associando um ou mais serviços a agentes no sistema; e v) agrupamento e síntese das competências com a finalidade de diminuir a codificação.

Para implementação foram utilizados agentes genéricos desenvolvidos em um projeto chamado OSACA (*Open System for Asynchronous Cognitive Agents*)(AZEVEDO H. ; SCALABRIN, 2005). Tais agentes possuem todos os mecanismos para o funcionamento no ambiente. A criação de novos agentes é feita fazendo-se clones de tais agentes e designando para eles tarefas no sistema.

Os agentes estão inseridos em um ambiente distribuído, podendo estar na máquina cliente ou em servidores na rede. Os agentes conseguem aprender através da comunicação, pois todas as mensagens são recebidas pelos agentes, que filtram as mensagens de acordo com sua destinação.

2.3.2 GRACILE (*Japanese Grammar Collaborative Intelligent Learning Environment*)

GRACILE (AYALA; YANO, 1997) é um ambiente de apoio à aprendizagem colaborativa baseado em agentes inteligentes. Este ambiente apóia o aprendizado de uma segunda língua (Japonês). O ambiente basicamente possui dois tipos de agentes: um agente chamado agente de domínio (*domain agent*) e outro tipo de agente denominado agente mediador (*mediator agent*). Os agentes de domínio ajudam os alunos no aprendizado usando algumas regras predefinidas internamente nele através da linguagem Prolog e verifica padrões da língua Japonesa através destas regras. O autor chama este conceito de elementos de conhecimento de domínio (*domain knowledge elements*). O agente mediador é aquele responsável pela colaboração no ambiente analisando meios de melhorar o processo de aprendizagem no ambiente.

¹O Portfólio, enquanto ferramenta pedagógica, pode ser descrito como uma coleção organizada e planejada de trabalhos produzidos pelo(s) aluno(s) ao longo de um determinado período de tempo (GONÇALVES, 2010)

2.3.3 I-Help

O I-Help (CAO; GREER, 2004) é um sistema on-line para aprendizagem colaborativa que oferece ajuda aos seus usuários na internet. Através dele, os estudantes compartilham seus conhecimentos entre si de forma colaborativa. Este sistema possui dois componentes principais: componente de discussão pública e componente de discussão privada. No componente de discussão pública, o estudante posta suas dúvidas e questões, que são gravadas na base de dados do I-Help. No componente de discussão privada, a discussão é efetuada em pares. No I-Help cada pessoa possui um agente ou assistente pessoal. Quando a pessoa tem um questionamento, o sistema irá localizar entre os demais a melhor pessoa que pode ajudá-lo na questão ou o melhor recurso que pode auxiliá-lo na resolução do problema.

O I-Help também permite que seus usuários possam configurar seus assistentes (agentes), personalizando-os com inclusão de regras. Dentro do sistema há algumas formas de fazer esta personalização. Uma delas é utilizando uma interface de gerenciamento de regras (*Agent Rule Management System*). Outra forma é através de uma interface para programação dos agentes em regras do CLIPS (DING; WANG; HUANG, 2009) que é um sistema para representação de conhecimento.

2.3.4 FLE (*Future Learning Environment*)

FLE (MUUKKONEN; HAKKARAINEN; LAKKALA, 1999) é um ambiente web para aprendizagem colaborativa baseada na construção de conhecimento (*collaborative knowledge building*). Em (MUUKKONEN; HAKKARAINEN; LAKKALA, 1999), os autores discutem um modelo pedagógico de construção progressiva do conhecimento (*Progressive Inquiry*), afirmando que o conhecimento não é gerado tão somente da resolução de problemas e explicação dos instrutores, mas sim de um processo de construção progressiva.

O ambiente possui os seguintes módulos: módulo WebTop, módulo de construção de conhecimento e módulo de Jamming. O WebTop é uma área onde os estudantes e professores podem armazenar arquivos pessoais. O módulo de construção de conhecimento é considerado o mais importante pois é nele que o processo pedagógico de construção progressiva de conhecimento é exercitado. O módulo de Jamming é um espaço compartilhado para construção colaborativa de artefatos digitais (figuras, textos, áudio ou vídeo).

Em (CHEN; WASSON, 2005), é feito um estudo de inclusão de agentes inteligentes no FLE. Um agente facilitador foi posto neste ambiente para efetuar algumas funções, como, por exemplo, monitorar as atividades de colaboração entre os estudantes, verificando a frequência de postagem de um determinado estudante no sistema. Outra função deste agente no ambiente é oferecer conselhos ao estudante durante o processo de construção do conhecimento, sem no entanto atrapalhar suas atividades (os conselhos aparecem em uma janela que não atrapalha os estudantes). Um agente foi projetado para

oferecer algumas informações, tais como: quem está on-line; informações de atualizações de módulos; e estatísticas de colaboração dos grupos. Os agentes ainda podem verificar as estatísticas de colaboração e enviar e-mails para os instrutores sobre alguma deficiência de participação de algum dos estudantes, além de alertar um determinado estudante sobre sua pouca participação.

O agente facilitador monitora o FLE coletando diversos dados e gravando em sua base de dados. O usuário tem a opção de obter informações do sistema através das atualizações e estatísticas geradas pelo agente em tempo real. Através de um componente chamado Gerador de Conselhos (*advice generator*), o agente gera os conselhos baseado em sua base de conhecimento. Além disso, o agente possui um aprendizado gerado pelo feedback dos usuários, permitindo incluir novas regras na sua base de conhecimento.

2.3.5 I-MINDS (*Intelligent Multiagent Infrastructure for Distributed System in Education*)

I-MINDS (SOH et al., 2006) (LIU et al., 2003) é um ambiente de aprendizagem que simula uma sala de aula. Este ambiente está estruturado em camadas e possui uma infra-estrutura hierárquica. No I-MINDS foram desenvolvidos três tipos de agentes inteligentes: agente professor, agente estudante e agente proxy remoto. O agente professor interage com o professor, e é responsável por: i) disseminar informações para os agentes estudantes ou agentes proxy remotos; ii) manter os perfis de todos os estudantes; iii) avaliar o progresso e a participação de diferentes estudantes; iv) gerar questionários diferenciados para os diversos estudantes; v) classificar as perguntas feitas pelos estudantes; e vi) gerenciar o progresso de uma sala de aula. O agente estudante interage com um estudante. Este agente guarda um perfil de outros estudantes e, caso o estudante necessite de ajuda para resolução de um problema, o agente irá verificar com outros agentes estudante alguém que possa ajudar este estudante na resolução de determinado problema. Caso não obtenha resposta, o agente irá contactar o agente professor para obter a ajuda necessária.

2.3.6 ICSS (*Intelligent Collaborative Support System*)

O ICSS (ISRAEL; AIKEN, 2007) apóia o trabalho colaborativo entre pequenos grupos de estudantes. Este ambiente está embasado em outro trabalho feito anteriormente o *Intelligent Collaborative Learning System* (ICLS). O objetivo do ICSS é guiar o grupo para atingir a colaboração máxima. Os agentes neste ambiente são orientados a metas, e são cooperativos para alcançar os objetivos. Os agentes são capazes de executar uma determinada ação no ambiente, que pode ser uma mensagem para o usuário, uma classificação de uma comunicação, entre outras.

No ICSS existe um componente chamado Líder de Grupo (Group Leader), que é um mecanismo

que oferece a interface entre os estudantes e os agentes. Este mecanismo irá receber entradas dos estudantes e repassar aos agentes do sistema, além de monitorar a colaboração entre os integrantes de um grupo de estudantes. Para cada estudante é desenvolvido um modelo (modelo de estudante), o qual é guardado pelo Líder de grupo. Este modelo é construído baseado em diversos parâmetros de avaliação. O agente, baseado em uma função de avaliação, atribui uma média ao estudante, que é usada para avaliar sua colaboração no grupo.

Além do modelo do estudante, é elaborado um modelo do grupo, que também reflete o perfil de determinado grupo no ambiente. Os modelos são atualizados pelo líder de grupo baseado em informações recebidas dos agentes no sistema. A interação entre os estudantes e o grupo são efetuadas pelo líder de grupo com base nestes modelos.

Capítulo 3

Aprendizagem Colaborativa e as Teorias de Aprendizagem

Este capítulo descreve aspectos relacionados à aprendizagem colaborativa e às teorias de aprendizagem que apóiam esta abordagem. A Seção 3.1 esclarece algumas definições sobre a Aprendizagem Colaborativa e a Aprendizagem Colaborativa Apoiada por Computador. A Seção 3.2 apresenta diversas teorias de aprendizagem, tendo como foco a PBL. Ao final desta seção, na Subseção 3.2.10, são mostrados alguns ambientes de aprendizagem que têm como base alguma teoria de aprendizagem em particular e na Subseção 3.2.11 é mostrado o uso de ontologias em ambientes de aprendizagem..

3.1 Aprendizagem Colaborativa

Aprendizagem Colaborativa é definida como aprendizado por colaboração (DILLENBOURG, 1999). Está explícito no termo que este tipo de aprendizagem surge da colaboração entre os estudantes para resolução de problemas comuns. Esta perspectiva nos mostra a importância da formação de grupos colaborativos por estudantes que possuem um objetivo comum.

Neste modelo de aprendizagem, há uma contribuição de cada membro do grupo, tendo portanto uma manutenção da contribuição individual. O indivíduo, posteriormente, poderá ser avaliado de forma individual. A aprendizagem se dá em um ambiente onde todos os membros têm o mesmo grau, ou seja, não há hierarquias. Cada membro participa da resolução do problema motivado por seu sucesso individual, pelo menos inicialmente, e o professor surge como um estimulador e orientador deste processo de aprendizagem (GONZÁLEZ, 2005).

O conhecimento é adquirido por meio da interação entre os participantes. O processo de ensino deixa de seguir um modelo tradicional, onde há apenas uma transmissão de informações unidirecional

(do professor para o estudante), e desenvolve a discussão e o pensamento crítico no estudante. Assim, o professor tem um papel de facilitador deste processo, deixando de ser um mero transmissor de informações para os estudantes. A aprendizagem é auto-dirigida, ou seja, o estudante é responsável pelo seu próprio aprendizado. Além disso, o estudante possui a responsabilidade perante o grupo de cumprir as tarefas para a solução do problema proposto.

Segundo (NIQUINI; BOTELHO, 1999) citado por (MENDES NETO, 2000), criar problemas (problematizar) e colocar o estudante em constante situação desafiadora são tarefas do facilitador, enquanto encontrar soluções e superar os desafios, a partir do já construído, faz parte do compromisso do estudante com sua própria aprendizagem.

Com a popularização dos cursos a distância e o crescimento das redes e internet, tornou-se comum o uso de computadores em rede para prover a aprendizagem colaborativa entre estudantes geograficamente distribuídos, esta abordagem é conhecida como Aprendizagem Colaborativa Apoiada por Computador (*Computer-Supported Collaborative Learning* - CSCL). Os sistemas que implementam a aprendizagem colaborativa desta forma precisam incluir mecanismos de cooperação e comunicação para que o processo de aprendizagem possa ser efetuado com qualidade. Além disso, tais sistemas devem possuir mecanismos para criação de grupos e formas de acompanhamento pelos facilitadores.

3.2 Teorias de Aprendizagem

Muitas teorias apóiam a aprendizagem colaborativa. Esta seção tem por objetivo apresentar as principais teorias de aprendizagem que oferecem apoio a esta modalidade de aprendizagem.

3.2.1 Construtivismo Piagetiano

Um dos principais estudiosos desta corrente, Jean Piaget, mostra em seus estudos como o conhecimento é adquirido pelo indivíduo. Ele tenta entender como o indivíduo passa de uma fase de menor conhecimento para outra de maior, sendo sua principal obra a *epistemologia genética*.

Segundo esta teoria, o conhecimento não é algo inerente ao indivíduo, mas parte da relação entre o indivíduo e o meio em que ele está inserido. Neste caso, o indivíduo é um ser ativo que obtém conhecimento através da relação entre este e os objetos do meio em que ele está inserido.

Segundo Piaget (PIAGET, 1973) citado por (CASTAÑÓN, 2005) (CASTAÑÓN, 2007), existem estruturas cognitivas no ser humano que são capazes de armazenar novas informações extraídas do meio, as quais ele chama de esquemas. Assim, no processo de construção do conhecimento, dois conceitos são definidos: assimilação e acomodação.

Assimilação se refere ao processo dinâmico e contínuo de aquisição do conhecimento. Já a acomodação se refere ao fato de como os esquemas irão se reorganizar para acomodar este novo conhecimento que foi assimilado. Em outras palavras, quando uma criança ou qualquer pessoa tenta adquirir conhecimento ou passa por uma situação (experiência) nova, ela primeiramente tenta assimilar essa experiência em seus esquemas existentes (CASTAÑON, 2005) (CASTAÑON, 2007). A tendência é que o esquema se modifique de modo a acomodar-se a esta nova informação. Em seus estudos, Piaget trata das fases de evolução desses esquemas, do nascimento até a idade adulta. Ele define quatro estágios nesta evolução: i) sensório-motor, que constitui a fase exploratória da criança através de movimentos motores, para alcançar um objeto por exemplo, respondendo a estímulos do ambiente; ii) pré-operatório, onde a criança começa a criar símbolos mentais e realizar associações, podendo realizar associações mesmo na ausência do objeto concreto; iii) operatório concreto, onde a criança começa a pensar de maneira lógica e começa a entender melhor o meio e suas ações sobre ele; e iv) formal, que possui estruturas que possibilitam raciocínio complexo e é capaz de discutir teorias e problemas reais.

3.2.2 Teoria Socio-Cultural de Vygotsky

A teoria sociocultural de Vygotsky (VIGOTSKI, 1998) (RIBAS; MOURA, 2006) enfatiza que a inteligência do indivíduo se origina da sociedade e cultura, ou seja, de uma interação entre o indivíduo e o ambiente no qual ele está inserido. Esta teoria de aprendizagem enfatiza que o conhecimento humano advém primeiramente do meio, através de relações interpessoais, e não de aspectos internos (intrapessoal) do indivíduo. Esta perspectiva remete à importância da sociabilidade e interação das pessoas no desenvolvimento mental dos indivíduos. Este é o elemento fundamental da teoria de Vygotsky sobre interação social, ou seja, que esta possui um papel formador e construtor. Isto significa que muitas funções mentais do indivíduo não são possíveis de serem construídas sem as interações sociais.

Em (VIGOTSKI, 1998), é apresentado um dos principais conceitos desta teoria de aprendizagem, a Zona de Desenvolvimento Proximal - ZDP. Segundo este conceito, os indivíduos possuem uma capacidade de resolução de problemas elementar e outra capacidade que só pode ser alcançada com a ajuda de um tutor (facilitador). Este limite corresponde à ZDP. Assim, cada um é capaz de resolver alguns problemas sem ajuda e isto representa seu conhecimento em uma dada idade. Contudo, o indivíduo pode atingir um limite (ZDP) caso obtenha ajuda do facilitador no processo de aprendizagem, onde este limite seria sua idade mental real. Vygotsky analisou em sua obra exemplos de crianças em seu processo de aprendizagem inicial.

É importante salientar que neste aspecto o estudante possui uma grande importância no processo de aprendizagem e o tutor exerce um papel de facilitador da aprendizagem, podendo guiá-lo na resolução de problemas mais complexos, que o estudante por si só não poderia solucionar. Esta teoria pode ser facilmente aplicada em ambientes de aprendizagem colaborativa, pois, além das interações com os

demaís colegas no processo de aprendizagem, o facilitador desempenha o papel do tutor neste processo, dinamizando o processo de aprendizagem.

3.2.3 Cognição Distribuída

A Cognição Distribuída foi desenvolvida por Ed Hutchins (HUTCHINS, 1995) citado por (ROGERS, 2006), sendo considerada uma mudança radical na forma de pensar sobre o fenômeno da cognição. A visão tradicional enfoca que o fenômeno de cognição é intrínseco ao indivíduo, enquanto Hutchins afirma que este processo se dá de forma distribuída no ambiente, englobando fatores externos, sobretudo artefatos do meio (ROGERS, 2006).

Assim, a cognição distribuída está relacionada com a interação do indivíduo com os artefatos do meio e não só com uma perspectiva interna. Em (HUTCHINS, 1995) citado por (ROGERS, 2006), são mostrados alguns exemplos do uso desta teoria em problemas reais, como, por exemplo, a análise do processo de navegação de um navio e a análise de um piloto de avião em sua cabine. Em outro exemplo, Rogers (1997) mostra um estudo de como controladores aéreos interagem com um sistema de radar ao controlar o tráfego. Estes exemplos mostram que a cognição está relacionada com o indivíduo e sua relação com o meio.

3.2.4 Teoria da Flexibilidade Cognitiva

Esta teoria de aprendizagem foi proposta por Spiro na década de 80 (SPIRO et al., 2003). Esta teoria afirma que as pessoas adquirem conhecimento em ambientes não-estruturados (complexos) pela associação entre unidades conhecidas. Este conceito pode ser melhor ilustrado (implementado) em ambientes que utilizam documentos hipermídia, onde os documentos estão interrelacionados, ou seja, os conceitos similares estão dispostos através de referências cruzadas.

A teoria da flexibilidade cognitiva afirma que o aprendizado é dependente do contexto do aprendiz. A teoria está embasada em teorias construtivistas e também enfatiza a construção do conhecimento pelo indivíduo. Assim esta teoria é bem aplicada em ambientes interativos de aprendizagem, onde o usuário aprende pela interação com o ambiente de aprendizado.

Em (GREENE; ROGERS, 2006), foi descrito um framework para avaliação de alunos da área de saúde, baseado na teoria da Flexibilidade Cognitiva, com o objetivo de verificar se os alunos atingiram realmente as habilidades necessárias.

3.2.5 Teoria da Aprendizagem Significativa

Teoria proposta por Ausubel (AUSUBEL, 1963) citado por (RISSOLI; GIRAFFA; MARTINS, 2008), que defende que a aprendizagem se dá através de um conhecimento prévio do indivíduo, sendo que o novo conhecimento é adquirido através da conexão com este conhecimento prévio e estruturas cognitivas presentes no estudante. Assim, um conhecimento significativo é obtido da interconexão de conhecimentos prévios que já estão estruturados na mente do aprendiz. Esta forma de aquisição de conhecimento parte de duas hipóteses fundamentais: a primeira é que as pessoas em geral aprendem partindo de um conceito mais geral e, a partir daí, assimilam melhor as partes específicas; a segunda é que a organização do conhecimento na mente ocorre de uma forma hierárquica, ou seja, conceitos mais gerais estão no topo e os mais específicos são incorporados posteriormente a estes conceitos prévios (RISSOLI; GIRAFFA; MARTINS, 2008).

Ausubel denomina os aspectos relevantes, nos quais serão agregados os demais conhecimentos, de conceito subsunçor (AUSUBEL, 1963). O conceito subsunçor é a base para agregação dos demais conhecimentos, sendo que a mente é formada por uma estrutura hierárquica com diversos conceitos subsunçores. Existe, no entanto, uma situação onde o aprendiz se depara com um conhecimento totalmente novo, sendo necessário formar um novo conceito subsunçor. Este novo conceito é formado a partir de estruturas denominadas organizadores prévios, que manipulam as estruturas cognitivas para estruturar o novo conceito na mente.

3.2.6 Aprendizagem Baseada em Problemas - *Problem-Based Learning*

A aprendizagem baseada em problemas (*Problem-Based Learning* – PBL), segundo (HMELO-SILVER, 2004), é um método no qual os estudantes aprendem através da resolução de um problema, que em geral não possui uma solução trivial e uma única solução correta. A aprendizagem é centrada no estudante e auto-dirigida. Os estudantes trabalham em pequenos grupos colaborativos para identificar o que eles necessitam aprender para resolução do problema. O professor atua como facilitador do processo de aprendizagem ao invés de apenas passar conhecimentos.

Na abordagem tradicional de ensino, primeiramente são apresentados os materiais de apoio e, em seguida, são feitas avaliações em cima destes materiais e aulas que foram lecionadas. Na PBL, primeiro é apresentado o problema e os estudantes necessitam procurar materiais e formas de resolvê-lo. O método de avaliação também é diferente, pois não consiste em provas, mas na avaliação do desenvolvimento das habilidades e conhecimento.

O facilitador tem o papel de guiar os estudantes neste processo, identificando deficiências de habilidades necessárias à solução do problema proposto. Assim, nesta teoria de aprendizagem, ao invés do

facilitador apenas repassar os conhecimentos e depois testá-los através de testes, ele faz com que os estudantes apliquem o seu conhecimento em situações novas. Os estudantes se deparam com problemas muitas vezes mal estruturados e tentam descobrir, através da investigação e pesquisa, soluções úteis.

3.2.6.1 Ciclo de Desenvolvimento da PBL

A PBL está estruturada em um ciclo de desenvolvimento contendo as seguintes etapas:

- É apresentado um problema ao estudante, chamado cenário do problema;
- Os estudantes formulam e analisam o problema e extraem fatos relevantes ao problema em questão;
- Estudantes têm um melhor entendimento do problema e formulam hipóteses para uma possível solução;
- O conhecimento adquirido é avaliado e são identificadas deficiências neste conhecimento;
- Um novo conhecimento é gerado a partir desta avaliação anterior;
- Ao final de cada problema, os estudantes irão refletir sobre os conhecimentos adquiridos.

A Figura 3.1 ilustra o ciclo de desenvolvimento da PBL.

3.2.6.2 Metas da PBL

A teoria de ensino-aprendizagem da PBL possui várias metas em relação ao estudante. Em (HMELO-SILVER, 2004), são elencadas as seguintes metas para os estudantes:

- Construir uma base de conhecimento flexível e extensível;
- Desenvolver habilidades efetivas de resolução de problemas;
- Desenvolver habilidades de aprendizagem auto-dirigida;
- Tornar-se colaboradores efetivos;
- Motivar intrinsecamente o estudante a aprender.

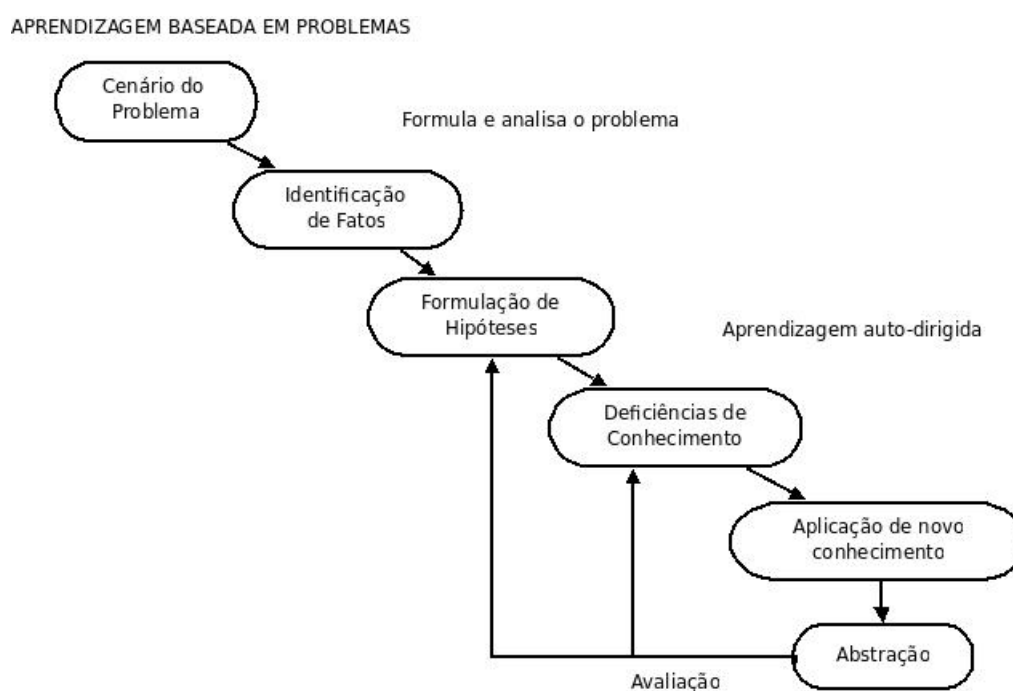


Figura 3.1: Ciclo da PBL adaptado de (HMELO-SILVER, 2004)

3.2.6.3 O Problema na PBL

O problema na PBL desempenha um papel fundamental no processo de aprendizagem. Este problema deve possuir algumas características particulares, entre as quais destacam-se: i) precisa ser multidisciplinar, ou seja, necessita envolver várias áreas de conhecimento; ii) precisa ser mal estruturado e complexo, com o objetivo de instigar os estudantes a levantar questionamentos que posteriormente irão fazer com que eles busquem conhecimentos que em um primeiro momento eles não possuíam; iii) o problema deve exigir um trabalho em equipe e motivar a colaboração entre os membros do grupo, ou seja, ele precisa ter um formato tal que motive esta socialização e divisão de tarefas; iv) precisa instigar a investigação e elaboração de hipóteses através das idéias; e, por fim v) precisa motivar cada estudante em particular ao aprendizado e resolução de problemas.

3.2.6.4 O Papel do Facilitador na PBL

O facilitador desempenha um papel primordial na PBL. Ele irá modelar boas estratégias de aprendizado, que deverão ser compostas de vários questionamentos, de modo a motivar o raciocínio dos estudantes. Ele possui também a responsabilidade de guiar os estudantes em torno dos vários estágios da PBL, bem como realizar o monitoramento do progresso do grupo e dos estudantes em particular. Além disso, o facilitador poderá se dirigir a um estudante em particular e verificar, através de questionamentos, se ele está realmente entendendo e participando da solução do problema. Ele precisa saber se existem

deficiências de habilidades necessárias ao cumprimento das etapas e usar estratégias apropriadas para correção destas deficiências e estímulo dos estudantes.

3.2.7 Comparativo das Teorias de Aprendizagem

A Tabela 3.1 mostra um quadro comparativo das diversas teorias de aprendizagem apresentadas anteriormente. Os parâmetros utilizados foram: **Aquisição de Conhecimento**, ou seja, a forma como a teoria explica a origem do conhecimento no indivíduo; o **Foco** da teoria, ou seja, o que ela enfoca como princípio da aprendizagem; e a **Aplicação** da teoria, ou seja, onde ela é melhor aplicada.

Tabela 3.1: Quadro comparativo das teorias de aprendizagem

Teoria	Aquisição de Conhecimento	Foco	Aplicação
Construtivismo Piagetiano	Interação com o ambiente (indivíduos e objetos)	Na interação	Estudos de desenvolvimento inicial com crianças
Teoria Socio-Cultural de Vygotsky	Interação social	Colaboração e sociabilidade	Ambientes colaborativos
Cognição Distribuída	Interação com artefatos (ex. sistemas de navegação de aeronaves)	Indivíduos e artefatos	Estudos de interação entre indivíduo e artefatos
Flexibilidade Cognitiva	Interação com ambientes hipermídia	Ambientes hipermídia	Ambientes interativos de aprendizagem
Aprendizagem Significativa	Com base em conhecimentos prévios do indivíduo	Conhecimentos prévios	Ambientes tutoriais
Aprendizagem Baseada em Problemas	Resolução de problemas	Problema	Centros de formação e ambientes colaborativos

3.2.8 Por que PBL?

Esta subseção tem por objetivo justificar a escolha pela PBL. Além das vantagens já apresentadas da aplicação desta teoria, nossa escolha foi embasada nos fatos a seguir:

- A PBL tem sido aplicada com sucesso em diversos estudos (LAI, 2005), (CARLISLE; IBBOTSON, 2005), (YEO; TAN; LEE, 2006), (O'KELLY; GIBSON, 2006), (TSENG; CHIANG; HSU, 2008);

- Estudos confirmam sua eficácia em relação a outros métodos de ensino (HMELO-SILVER; BARROWS, 2006), (STROBEL; BARNEVELD, 2009);
- A PBL já foi aplicada com sucesso em ambientes virtuais de aprendizagem (YUEH; LIN, 2005), (SUEBNUKARN; HADDAWY, 2007), (ŞENDAĞ; ODABAŞI, 2009).

Além disso, a PBL é adequada à aprendizagem colaborativa apoiada por computador pois exige trabalho em grupo e colaboração, aumenta a motivação dos estudantes e desenvolve sua capacidade de resolução de problemas.

3.2.9 Requisitos de Ambientes CSCL para aplicação bem sucedida da PBL

Um ambiente web para apoiar a aprendizagem colaborativa, usando a PBL como estratégia pedagógica, deve oferecer ferramentas para apoiar os três principais estágios da solução do problema: i) o *brainstorming* inicial; ii) geração de idéias para resolução do problema; e iii) as explicações e justificativas.

O *brainstorming* inicial pode ser efetuado através das ferramentas de comunicação síncrona, que são baseadas em comunicação em tempo real. Estas ferramentas promovem a eficácia do processo, oferecendo feedback imediato. A geração de idéias para solução do problema, tanto as idéias iniciais quanto as idéias formadas depois de pesquisas e leituras sobre o problema, podem ser efetuadas através de ferramentas de comunicação assíncrona. Neste caso, a interação entre facilitadores e estudantes não acontece em tempo real. As principais vantagens desta modalidade de comunicação são flexibilidade, tempo para pensar a respeito - ambos facilitador e estudantes têm oportunidade para amadurecer suas idéias e consultar fontes com antecedência - e custo razoável. Explicações e justificativas podem usar ferramentas de comunicação síncrona e assíncrona e ferramentas de cooperação (PUNTAMBEKAR, 1999; LUCENA et al., 2006).

Para o correto funcionamento do ambiente de apoio à aprendizagem colaborativa baseada na web são necessários também ferramentas de coordenação. Estas ferramentas correspondem a mecanismos de administração do ambiente e a coordenação das atividades oferecidas pelo ambiente (LUCENA et al., 2006). As ferramentas de coordenação possibilitam a administração de todos os outros mecanismos disponíveis no ambiente. Elas também possibilitam a realização de tarefas como controle de permissões para uso de recursos, registro de usuários, determinando o perfil deste, entre outras (LUCENA et al., 2006).

3.2.10 Sistemas Multiagentes e as Teorias de Aprendizagem

Como visto anteriormente, a PBL possui algumas dificuldades para ser implementada. Estes desafios se referem, principalmente, à sua implementação no ensino presencial. Na implementação em EaD existem mais alguns desafios, pois nem sempre o facilitador possui todas as informações necessárias para realizar o processo de tutoria e detectar possíveis problemas que possam ocorrer no processo de aprendizagem. Os SMAs podem auxiliar na implementação da PBL, oferecendo ao facilitador e aos estudantes subsídios para o cumprimento das etapas da PBL e resolução de problemas. Desta forma o ambiente de ensino-aprendizagem será mais pró-ativo, pois os agentes podem auxiliar no cumprimento da teoria no ambiente, provendo mecanismos que possam auxiliar estudantes e facilitadores, aumentando sua produtividade. Além disso, os ambientes interativos de aprendizagem possuem como característica a dinamicidade e os agentes são facilmente aplicados em ambientes assim, pois em geral atuam em ambientes complexos e dinâmicos. Alguns trabalhos que descrevem o uso de agentes em ambientes de aprendizagem são apresentados a seguir.

Em (BASTOS FILHO, 2006), é mostrado um protótipo de um sistema, chamado *Intelligent Open Challenges System* (IOCS), que está sendo desenvolvido com base no método clínico de Piaget. Segundo o autor, o sistema tem como desafio respeitar a individualidade do processo mental cognitivo de cada usuário, registrando suas ações e reações diante das intervenções dos agentes inteligentes na resolução de provas lógicas de concepção aberta, denominadas de Desafios Abertos.

Em (RIZZI, 2006), é descrita uma interpretação sobre a cooperação humana, particularmente sobre a cooperação na ação, no contexto da Epistemologia Genética, e seu emprego enquanto fundamento teórico para definir um conjunto de requisitos para agentes computacionais e sistemas multiagente.

Em (MOISIL et al., 2006), é descrito um modelo para ambientes virtuais de aprendizagem que emprega agentes inteligentes para implementar a teoria socio-cultural de Vygotsky, enfocando o aspecto social de interação. Este trabalho enfatiza o estudante como principal ator no processo de aprendizagem.

Em (RISSOLI; GIRAFFA; MARTINS, 2008), é descrita a proposta de um sistema tutor inteligente que está embasado na teoria da aprendizagem significativa de Ausubel, acrescido de um componente baseado em Lógica Fuzzy para monitoramento da evolução dos estudantes no sistema.

Em (AZEVEDO H. ; SCALABRIN, 2005), é descrito um ambiente multiagente de aprendizagem colaborativa que foi implementado seguindo os padrões da Aprendizagem Baseada em Projetos (*Project-Based Learning*).

3.2.11 Uso de Ontologias em Ambientes de Aprendizagem

Ontologias são formas de representar o conhecimento de um dado domínio. No contexto da web, elas são usadas principalmente para padronização e busca de recursos e para dar significado à web, facilitando a busca por informações na rede de uma forma eficiente. As ontologias podem ser usadas em ambientes de aprendizagem para representação de conceitos e modelos inerentes ao ambiente em questão. Por exemplo, em (BITTENCOURT et al., 2006), é descrita uma ontologia para construção de ambientes interativos de aprendizagem. São modelados, através de uma ontologia, diversos conceitos de um ambiente de aprendizagem, tais como: modelo do estudante, o qual representa o estudante que será ensinado; modelo pedagógico, que se refere às estratégias para aprendizagem; modelo de colaboração, que define a forma de colaboração no ambiente; e modelo de domínio, que representa as informações que serão relevantes ao ensino, ou seja, referente ao que será ensinado.

As ontologias são usadas para diversos propósitos em ambientes de aprendizagem, contudo uma grande aplicação desta tecnologia é na área de personalização do ambiente. Isto é feito levando em consideração características únicas de cada estudante. Este comportamento do ambiente é desejável, pois ele pode adaptar-se aos estilos de aprendizado de cada estudante. Em (GASCUENA; FERNANDEZ-CABALLERO; GONZALEZ, 2006), é elaborada uma ontologia de domínio para personalização de materiais em relação ao estilo de aprendizagem do estudante. Em (MIN; WEI; LEI, 2008), é proposto um framework para adaptação em ambientes de aprendizagem, neste caso, dois modelos ontológicos são construídos: um modelo de conhecimento e um modelo do aprendiz, sendo que no modelo do aprendiz é informado o objetivo do estudante e no modelo de conhecimento são gravadas informações relacionadas a este objetivo. Ainda no modelo do estudante, são armazenados estilos de aprendizado, atividades de aprendizado e avaliações. Portanto, com base no modelo do estudante construído através de uma ontologia, o sistema irá se adaptar de forma a cumprir os requisitos do estudante.

Em (GOMES et al., 2006), é mostrado um modelo usado para personalização do ambiente de acordo com o monitoramento do estudante. Inicialmente o estudante interage com o sistema através de uma interface, que posteriormente será adaptada de acordo com suas características particulares. Em (GLADUN et al., 2009), é mostrado o uso de ontologias de domínio para controlar e avaliar o estudante concernente à aquisição do conhecimento. Assim, uma ontologia de domínio é construída pelo próprio estudante e esta é comparada com outra de referência. Através dos erros encontrados, é possível avaliar o nível de aquisição do conhecimento pelo estudante.

Capítulo 4

Metodologias para Modelagem de Sistemas Multiagente: um Estudo Comparativo

Este capítulo tem por objetivo apresentar diversas metodologias de modelagem de sistemas multiagente, e um estudo comparativo entre elas. A partir deste estudo comparativo foi selecionada uma metodologia para ser usada na modelagem dos agentes que comporão a arquitetura de agentes de apoio à aprendizagem baseada em problemas.

4.1 Metodologias para Modelagem de Sistemas Multiagente

No processo de desenvolvimento de software faz-se necessário o uso de metodologias de apoio ao desenvolvimento. Uma metodologia pode ser definida como o processo usado na produção de software que emprega um conjunto de técnicas predefinidas e convenções, sendo assim, esta se apresenta como uma série de passos com técnicas associadas a cada passo. Tais passos geralmente são organizados em um ciclo de vida nas várias fases do desenvolvimento (RUMBAUGH et al., 1991).

A metodologia que pode ser seguida depende do paradigma de desenvolvimento utilizado. Por exemplo, para o paradigma orientado a objetos, existem as diversas metodologias orientadas a objeto. A complexidade e a natureza distribuída e dinâmica das novas aplicações deram origem a um novo paradigma: o paradigma orientado a agentes. O surgimento deste paradigma ocasionou a pesquisa e desenvolvimento de novas metodologias de apoio ao desenvolvimento, cujas principais serão discutidas

neste capítulo.

O estudo das diversas metodologias orientadas a agentes foi motivado pela escolha da metodologia que se aplica ao desenvolvimento de agentes que poderão ser incluídos na arquitetura de agentes proposta neste trabalho. Ao final deste capítulo, as metodologias serão comparadas com base em parâmetros pré-definidos, tais como: completude da metodologia (quais fases do ciclo de vida do software a metodologia contempla?), documentação existente, consolidação da metodologia, adaptação ao problema de agentes no contexto de ambientes de aprendizagem (a metodologia é de propósito geral ou específico?), suporte de ferramentas de apoio ao processo da metodologia e, por fim, aplicabilidade da metodologia (seu uso na prática).

As metodologias que serão comparadas neste estudo são: MAS-CommonKADS, K-Org, Tropos, PASSI, Prometheus, Gaia, ADELFE, MESSAGE, INGENIAS e MaSE. Cada metodologia será descrita com foco em seus pontos principais, portanto sem entrar em detalhes específicos de implementação desta.

4.2 CommonKADS

O CommonKADS¹ (SCHREIBER, 2000) é uma metodologia para desenvolvimento de sistemas baseados em conhecimento. Existe um conjunto de modelos que compõem a metodologia CommonKADS, os quais são listados a seguir:

Modelo de Organização: este modelo analisa as principais características da organização, com vistas a descobrir os problemas e as possíveis aplicação de um sistema de conhecimento.

Modelo de Tarefas: o modelo de tarefas analisa as tarefas efetuadas na organização de forma global, suas regras, critérios para execução e aspectos relevantes às competências.

Modelo de Agentes: este modelo analisa os diversos agentes que atuam na execução das tarefas da organização, sejam eles pessoas ou sistemas computacionais, analisando suas competências, autoridade e influência na organização.

Modelo de Conhecimento ou Experiência: consiste em um dos principais modelos do CommonKADS. Neste modelo serão construídas as estruturas do conhecimento para execução de uma tarefa organizacional. Em outras palavras, é a construção da base de conhecimento da organização. Esta base será utilizada para as inferências do sistema construído.

¹A metodologia CommonKADS não é uma metodologia para modelagem de sistemas orientados a agentes inteligentes propriamente dita. Contudo, como algumas metodologias de modelagem de sistemas multiagente são embasadas no CommonKADS, faz-se necessário uma breve descrição desta metodologia. Vale salientar que o CommonKADS não é incluído no estudo comparativo.

Modelo de Comunicação: este modelo implementa a comunicação entre os diversos agentes envolvidos no sistema.

Modelo de Projeto: este último modelo representa a implementação do sistema, em termos de arquitetura, plataformas e métodos a serem utilizados na solução computacional.

Os três primeiros modelos (organização, tarefas e agentes) fazem parte da fase de análise da organização, ou seja, é o primeiro passo no desenvolvimento da metodologia e representa o entendimento do domínio, oferecendo a base para as demais fases. Os modelos de conhecimento e comunicação fazem parte da fase de descrição conceitual do sistema, bem como dos dados que serão manipulados pelo sistema. O modelo de projeto, por sua vez, realiza a implementação dos conceitos em um sistema computacional.

4.3 MAS-CommonKADS

A metodologia MAS-CommonKADS (IGLESIAS, 1998; IGLESIAS C.A. E GARIJO, 2005) é uma extensão da metodologia CommonKADS com abordagem voltada para modelagem de sistemas orientados a agentes.

A metodologia MAS-CommonKADS é dividida nas seguintes fases: conceituação, análise, projeto, codificação, integração, operação e manutenção. Assim como o CommonKADS, o MAS-CommonKADS também possui um conjunto de modelos que são executados ao longo de suas fases. Seus modelos são descritos a seguir:

Modelo de Agentes: especifica as características dos agentes, capacidades de raciocínio, sensores e atuadores, habilidades dos agentes e os grupos aos quais pertence cada agente.

Modelo de Tarefas: descreve as tarefas de cada agente, metas, métodos de solução de problemas, etc.

Modelo de Experiência ou Conhecimento: descreve o conhecimento necessário para cada agente com vistas ao cumprimento dos seus objetivos.

Modelo de Organização: descreve a organização na qual os agentes estarão inseridos, ou seja, descreve a relação da organização com seu meio.

Modelo de Coordenação: descreve os métodos de comunicação entre os agentes e suas interações.

Modelo de Comunicação: modela as interações entre as pessoas e o sistema. Esta parte refere-se ao desenvolvimento de interfaces.

Modelo de Projeto: se embasa nos modelos anteriores e se subdivide em três modelos: i) projeto

da rede, que desenvolve aspectos relevantes da infraestrutura de rede dos agentes; ii) projeto do agente, que seleciona a arquitetura mais apropriada para cada agente; e por último, iii) projeto de plataforma, que seleciona a plataforma mais apropriada para cada arquitetura de agente previamente escolhida.

A primeira fase da metodologia MAS-CommonKADS, a conceitualização, consiste na identificação dos requisitos funcionais do sistema. Nesta fase nenhum modelo descrito acima é usado, contudo casos de uso são utilizados para identificar os possíveis usuários do sistema e possíveis metas dos usuários, descrevendo-se as maneiras de se atingir tais metas. Duas técnicas são usadas nesta fase: UER (*User-Environment-Responsibility*), que analisa o sistema na perspectiva do usuário, do ambiente e em termos de requisitos que não necessitam da interação do usuário; e CRC (*Class-Responsibility-Collaboration*), que consiste em tabelas que definem, para cada agente, suas metas e planos, bem como o conhecimento necessário para atingir estas metas.

A segunda fase, de análise, consiste na especificação dos requisitos do sistema. Nesta fase são utilizados os seguintes modelos: agentes, tarefas, coordenação, conhecimento ou experiência e organização.

A última fase, projeto, trata da seleção da plataforma mais apropriada para o desenvolvimento do sistema multiagente, usando os requisitos coletados nas demais fases para implementação do sistema.

4.4 K-Org

K-Org (FALCÃO, 2002) é uma metodologia para modelagem de conhecimento organizacional (memória organizacional). Capacita as organizações a estruturar, representar, localizar, utilizar e desenvolver seu conhecimento coletivo para utilização para um fim proveitoso para a organização.

A metodologia K-Org auxilia na gerência de conhecimento pela organização. Esta gerência permite à organização tomar decisões em mudanças de mercado e atuar de forma competitiva.

A metodologia pretende captar o conhecimento da organização distribuído de forma heterogênea e de difícil padronização. Esta tarefa pode ser efetuada por um engenheiro de conhecimento, contudo nesta metodologia esta tarefa é efetuada por agentes inteligentes denominados neste contexto de K-Agts, que assumirão o papel do engenheiro de conhecimento na construção da memória organizacional.

A metodologia K-Org é dividida nas seguintes fases:

Fase Contextual: corresponde ao entendimento do amplo contexto organizacional, permitindo verificar oportunidades e aplicação de conhecimento neste contexto. Nesta fase é usado o Modelo de Organização e o K-agt da organização é especificado com o objetivo de entender a organização e alimentar a sua base de conhecimento. O modelo de tarefas também é confeccionado, e o K-agt de tarefas é projetado com o objetivo de entender as atividades na busca dos objetivos e alimentar sua base de conhecimento. O modelo de agentes é implementado, e o K-agt de atores é projetado com o objetivo

de entender os papéis e competências que os vários agentes possuem na organização para executar uma tarefa, alimentando sua base.

Fase Conceitual: busca a modelagem dos conceitos e estruturas que compõem o conhecimento da organização e da forma de comunicação entre os agentes envolvidos. O modelo de conhecimento é confeccionado e o K-agt de domínio é projetado, utilizando técnicas de representação do conhecimento. O modelo de comunicações também é criado nesta fase.

Fase de Projeto: mapeia o conhecimento e as habilidades dentro da empresa. Está dividida em duas visões: a externa, do usuário especialista, e a interna, do engenheiro do conhecimento. Utiliza o k-map para mapear o conhecimento da organização.

Fase de Desenvolvimento e Criação: utiliza o mapa de conhecimento como referência para tornar explícito o conhecimento organizacional utilizado nos processos de negócios e na solução de problemas correspondentes.

Fase de Implantação e Disseminação: publica o conteúdo da base de conhecimento de acordo com necessidades específicas.

Fase de Manutenção: através das atividades de gestão e evolução, utiliza uma base de conhecimento customizada para armazenar, indexar e recuperar o conhecimento explicitado na fase de desenvolvimento e criação, através das atividades de construção e testes. A estrutura do conhecimento é aquela definida na fase de projeto. A atividade de Suporte e Treinamento auxilia a organização a tornar explícito o conhecimento de forma colaborativa e continuada.

4.5 Tropos

Tropos (MYLOPOULOS, 2005; BRESCIANI et al., 2004) é uma metodologia orientada a agentes dirigida a requisitos. Esta metodologia usa conceitos de um framework de modelagem chamado *i** (YU, 1995).

Tropos possui quatro fases:

Análise de requisitos: esta fase divide-se em: análise de requisitos preliminar e análise de requisitos posterior. A fase da análise de requisitos preliminar consiste no entendimento da organização em termos de requisitos, ou seja, o que o sistema deve fazer para resolução das necessidades da organização. Na análise de requisitos posterior, o modelo conceitual desenvolvido inicialmente é expandido para modelar os requisitos funcionais e não-funcionais do sistema.

Em ambas as fases são usados dois tipos de diagramas: diagrama de atores, que é um diagrama que envolve as dependências entre os atores do sistema no que tange ao cumprimento de metas; e diagrama de raciocínio, que descreve o processo de raciocínio de cada ator no sistema.

Projeto da arquitetura: define a arquitetura global do sistema em termos de atores e suas dependências. Esta fase é efetuada em três passos: i) no primeiro passo toda a arquitetura do sistema é definida. Novos atores podem ser introduzidos, geralmente sub-atores; ii) o segundo passo é caracterizado pela definição das capacidades de cada ator no cumprimento de suas metas; e iii) o último passo consiste no agrupamento de atores de acordo com suas habilidades. Neste passo os agentes são associados em grupos conforme suas capacidades.

Projeto detalhado: nesta fase é feita uma especificação detalhada de cada agente. Metas, crenças e capacidades, bem como comunicação entre os agentes, são especificadas em detalhes. Esta fase se caracteriza pelas escolhas de implementação. Nesta fase são usados diagramas da UML, bem como AUML (ODELL; PARUNAK; BAUER, 2000), para definir protocolos específicos aos agentes. Estes diagramas são: Diagramas de Capacidade, que especificam a capacidade de um determinado agente; Diagramas de Planos, que modelam os planos a serem feitos; e Diagramas de Interação entre agentes, que usam AUML para modelagem da comunicação entre os agentes.

4.6 PASSI

PASSI (*Process for Agent Societies Specification and Implementation*) (COSSENTINO, 2005; BURRAFATO; COSSENTINO, 2002) é uma metodologia passo a passo que vai desde os requisitos ao código. Ela atua no desenvolvimento de sistemas multiagente usando conceitos extraídos da orientação a objetos usando UML.

A metodologia PASSI possui cinco modelos: modelo de requisitos de sistema, modelo de sociedade de agentes, modelo de implementação de agentes, modelo de código e modelo de implantação.

O **modelo de requisitos do sistema** tem por objetivo obter os requisitos do sistema e está dividido em quatro fases: Descrição de Requisitos de Domínio, contemplando uma descrição funcional do sistema usando casos de uso convencionais; Identificação de Agentes, onde as funções do sistema são agrupadas e distribuídas para cada agente; Identificação de Papéis, descrevendo uma sequência de diagramas relacionados a uma funcionalidade específica; e Especificação de Tarefas, especificando as capacidades de cada agente.

O **modelo de sociedade de agentes** é um modelo de interações e dependências entre os agentes envolvidos no sistema. Este modelo tem três fases: Descrição Ontológica, que usa diagramas de classes para descrever o conhecimento atribuído a cada agente e suas comunicações; Descrição de Papéis, onde diagramas de classes são usados para mostrar os papéis desempenhados por cada agente, as tarefas envolvidas, as capacidades de comunicação e as dependências entre agentes; e Descrição de Protocolos, que usa diagramas de sequência para especificar a gramática de cada protocolo de comunicação.

O **modelo de implementação dos agentes** é um modelo clássico de solução arquitetural baseado em classes e métodos. Ele se diferencia do método clássico por existir dois diferentes níveis de abstração, a saber: o nível social (múltiplos agentes) e o modelo de um único agente, que possui duas etapas: Definição de Estrutura de Agentes, onde diagramas de classe convencionais descrevem as classes dos agentes participantes da solução; e Descrição de Comportamento dos Agentes, onde diagramas de atividade descrevem o comportamento individual de cada agente.

O **modelo de código** envolve a geração de código base através de modelos do PASSI usando alguma ferramenta e complementação manual do código.

O **modelo de implantação** apóia a implantação do sistema. Isto envolve a atividade de testes em pelo menos dois níveis: testes de comportamento individual dos agentes e testes da sociedade de agentes.

A metodologia PASSI possui uma ferramenta que auxilia no processo de desenvolvimento da metodologia, o PASSI ToolKit (PTK) (COSSENTINO; LOMBARDO; SABATUCCI, 2010).

4.7 Prometheus

Prometheus (PADGHAM; WINIKOFF, 2004; PADGHAM L. WINIKOFF, 2005) é uma metodologia usada em aplicações industriais e para ensino de graduação. A metodologia se propõe a ser prática e detalhada.

A motivação da construção de Prometheus deve-se a diversos fatores, dentre os quais destacam-se: i) desenvolver uma metodologia prática, necessitando para tal ser precisa, completa e detalhada; ii) desenvolver agentes orientados a metas e planos; iii) facilitar a construção de ferramentas de modelagem que aplicam a metodologia; e iv) ser utilizada em diversas áreas, seja ela comercial ou de ensino.

A metodologia Prometheus é de uso geral e está baseada no modelo BDI (*Belief Desire Intention*), mas apesar de estar embasada neste modelo permite o desenvolvimento de agentes através de frameworks que não seguem este modelo.

Prometheus pode ser dividida de forma macro em três atividades: especificação de sistema, projeto da arquitetura e projeto detalhado.

A etapa de especificação de sistema parte da execução de algumas atividades: Metas do Sistema, onde é descrito o sistema em termos de metas em alto nível, gerando um artefato chamado *diagrama de metas*; Funcionalidades, que definem as capacidades do sistema, agrupando um conjunto de funcionalidades; Cenários, que são usados para descrever os passos a serem efetuados dado um evento; e Interface do Sistema, que descreve percepções e ações a serem efetuadas pelos agentes do sistema.

A etapa de projeto de arquitetura realiza três atividades: i) decidir os tipos de agentes que comporão o sistema, onde os agentes são identificados baseados nos grupos de funcionalidade; ii) descrever as interações entre os agentes no sistema através de diagramas de interação e protocolos de interação; e iii) Desenvolver arquitetura global do sistema, onde o sistema é descrito através de um diagrama de sistema panorâmico, que captura os tipos de agentes do sistema e os limites do sistema em termos de percepções e ações.

A etapa de projeto detalhado consiste em: i) desenvolver o interior dos agentes em termos de capacidades; ii) desenvolver diagramas de processo dos protocolos de interação; e iii) desenvolver outras capacidades que podem ser originadas de capacidades desenvolvidas anteriormente, bem como em termos de eventos, planos e dados. Capacidade é um mecanismo estruturado semelhante aos módulos, podendo conter: planos, dados ou eventos.

Prometheus possui uma ferramenta que auxilia no processo de desenvolvimento chamada PDT (Prometheus Design Tool) (PADGHAM; THANGARAJAH; WINIKOFF, 2005). Outra ferramenta que suporta a metodologia Prometheus é o Ambiente de desenvolvimento JACK (JDE) (BUSETTA et al., 1999).

Prometheus é uma metodologia que se propõe a ser prática e foi aplicada com sucesso em turmas de estudantes de graduação e em aplicações industriais. A metodologia traz em si modelos que são úteis ao desenvolvimnto prático.

4.8 Gaia

Gaia (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003, 2005) é uma metodologia que auxilia no desenvolvimento de sistemas multiagente simulando uma organização do mundo real. Esta metodologia é descrita em (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003, 2005) como metáfora organizacional.

Esta abordagem facilita a aproximação entre o sistema computacional e o mundo real e é ideal para modelagem de organizações. Gaia é uma metodologia orientada a papéis e, segundo este raciocínio, o sistema deve identificar todos os papéis na organização antes de iniciar o processo de modelagem. Gaia se embasa em um conjunto de abstrações, a saber: i) *ambiente*, que significa detectar no ambiente todos os recursos e entidades que o sistema pode usufruir para alcançar seus objetivos; ii) *papéis*, que especificam o que deve ser feito pelo agente na organização, ou seja, determinam a posição exata do agente na organização e o comportamento esperado; iii) *interações*, que definem como os papéis na organização interagem; iv) *papéis organizacionais*, que definem as competências e capacidades da organização em termos de objetivos a serem alcançados; e v) *estruturas organizacionais*, que definem a estrutura organizacional em termos das atividades organizacionais. A definição de tais abstrações permitem um melhor entendimento da metodologia.

O processo de desenvolvimento Gaia contém três fases: análise, projeto da arquitetura e projeto detalhado. A metodologia tem início na fase de análise, visando coletar e organizar a especificação que será base para a fase de projeto de arquitetura.

A **fase de análise** tem por objetivo entender o sistema e decompô-lo em papéis que serão desempenhados na organização, através do modelo de papéis, além de definir como os mesmos interagem, através do modelo de interação. Portanto, o modelo de papéis identifica os papéis existentes no sistema e o modelo de interação especifica um conjunto de definições de protocolos, um para cada tipo de interação entre os papéis. Esta fase de análise inclui identificar: a) as metas de agentes presentes no sistema e o comportamento esperado dos mesmos; b) o ambiente; c) os papéis iniciais; d) as interações iniciais; e e) as regras que os agentes devem seguir.

Os artefatos gerados na fase de análise são utilizados como entrada para a fase seguinte, ou seja, para o projeto de arquitetura. A fase da elaboração da arquitetura inclui a definição da estrutura organizacional do sistema em termos de sua topologia e regime de controle e a identificação completa dos papéis e interações.

A fase de projeto detalhado compreende a definição do modelo de agentes e a definição do modelo dos serviços que os agentes devem oferecer para desempenhar seus papéis. O modelo de agentes identifica as classes de agentes existentes no sistema, enquanto o modelo de serviços identifica os serviços (funções do agente) associados a cada classe de agentes e define as principais propriedades desses serviços.

4.9 ADELFE

ADELFE (Atelier de Développement de Logiciels à Fonctionnalité Emergente) (BERNON et al., 2005, 2002), que traduzido significa “conjunto de ferramentas para desenvolvimento de software com funcionalidades emergentes”, é uma metodologia para desenvolvimento de sistemas adaptativos usando agentes.

Um sistema adaptativo é definido como aquele que é capaz de modificar seu comportamento com base em determinados aspectos de sua experiência ou ambiente. ADELFE lida com agentes inteligentes em tais sistemas, através de um processo de cooperação entre eles com vistas à solução do problema.

ADELFE é baseado no *Rational Unified Process* (RUP) com adição de algumas particularidades referentes a sistemas adaptativos, tais como a caracterização do ambiente, a identificação de falhas de cooperação, etc. O uso do RUP implica o uso da UML e, para modelar os protocolos de interação entre os agentes, o ADELFE faz o uso da AUML (ODELL; PARUNAK; BAUER, 2000). Existem pelo menos três ferramentas integradas ao conjunto de ferramentas do ADELFE: i) uma ferramenta para checar se

o sistema necessita de uma solução adaptativa; ii) *OpenTool*, uma ferramenta de modelagem gráfica que suporta UML e foi modificada para suportar AUML²; iii) uma ferramenta interativa que descreve o processo e como aplicá-lo (BERNON et al., 2004).

As fases do ADELFE podem ser resumidas em: Requisitos Preliminares e Finais; Análise e Projeto. O ADELFE usa a mesma nomenclatura do RUP e as fases descritas anteriormente são chamadas de WD1, WD2, WD3 e WD4, respectivamente. Uma fase se subdivide em atividades (Ai), e uma atividade se subdivide em passos (Si). Aqui será mostrado apenas os itens que foram adicionados ao RUP para suportar o desenvolvendo de sistemas adaptativos.

Na fase de requisitos preliminares (WD1) nada mudou. Esta fase serve para levantamento dos requisitos e documentação dos mesmos. Na fase de requisitos finais (WD2), existe uma atividade importante que é a caracterização do ambiente (A6), que é feita em três passos: i) identificação das entidades que interagem com o sistema e a limitação destas interações (A6-S1); ii) no segundo passo (A6-S2), o contexto é estudado, através das interações entre as entidades e o sistema; e iii) finalmente, no terceiro passo (A6-S3), o ambiente é caracterizado segundo (RUSSELL; NORVIG, 2003) em: totalmente ou parcialmente observável, contínuo ou discreto, determinístico ou não-determinístico, dinâmico ou estático. Ainda em WD2, na atividade de determinação de casos de uso, possíveis falhas na cooperação devem ser modeladas como casos de uso (A7-S2).

Na etapa de Análise (WD3) foi acrescida a adequação ao uso da tecnologia de sistemas adaptativos com agentes inteligentes (A11). Esta análise é feita em dois passos (A11-S1 e A11-S2): o primeiro verifica a adequação global do sistema à tecnologia e o segundo verifica componentes internos ao sistema. Outra atividade que foi acrescida ainda na análise foi a identificação dos agentes (A12). Esta atividade é executada em três passos (A12-S1, A12-S2 e A12-S3): estudo das entidades no contexto do domínio, identificação das entidades cooperativas em potencial; e, finalmente, definição dos agentes em termos destas classes. Na atividade A13 de estudo das interações entre as entidades, foi incluído o passo S3 para estudar os relacionamentos entre os agentes.

Na última fase, projeto (WD4), foi incluída a atividade A15, que é um estudo das linguagens de interação para verificar como os agentes interagem. A próxima atividade (A16) também foi incluída e refere-se ao projeto de agentes, sendo executada em seis passos: i) projeto de habilidades (A16-S1), que define as habilidades dos agentes; ii) projeto de aptidões (A16-S2), que fornece as aptidões dos agentes; iii) projeto de linguagens de interação (A16-S3), que escolhe os protocolos com base no estudo de adequação destes; iv) projeto de representações (A16-S4), que especifica atributos e métodos que permitem ao agente criar uma representação de si mesmo, do ambiente ou de outros agentes; v) projeto de características (A16-S5), onde são descritas as propriedades intrínsecas ou físicas de um agente; e vi) projeto de situações não cooperativas (A16-S6), que consiste em um dos passos mais importantes, onde são definidas, através de modelos, situações não cooperativas do sistema.

²Esta modificação é descrita em (BERNON et al., 2004).

A última atividade incluída foi a prototipação rápida (A17), onde é efetuado um teste se os modelos correspondem ao esperado. Esta funcionalidade é garantida pelo *OpenTool*, que permite realizar uma simulação através dos modelos criados.

A metodologia ADELFE portanto é específica ao desenvolvimento de sistemas multiagente adaptativos, não se tratando de uma metodologia de uso geral.

4.10 MESSAGE

MESSAGE (GARIJO et al., 2005) é uma metodologia para modelagem de sistemas multiagente baseada no RUP e que abrange as fases de análise e projeto no ciclo de vida do desenvolvimento de software. MESSAGE estende a UML com novos conceitos relacionados a agentes e notações gráficas e enriquece e adapta o modelo de processo RUP para a análise e projeto de sistemas multiagente (GARIJO et al., 2005).

A etapa da análise serve para produzir uma especificação do sistema ou modelo de análise. O processo de análise é feito por um refinamento por etapas. No topo da análise ou no nível 0 é feita uma definição global do sistema. No próximo nível, o comportamento das entidades, tais como agentes, organização e objetivos, é definido. Outros refinamentos podem tratar de requisitos funcionais e não-funcionais.

O propósito do projeto é definir as entidades computacionais que representam o sistema multi-agente representado na fase de análise. Estes artefatos são transformados para possibilitar a implementação. Esta fase é dividida em dois processos: projeto de alto nível e projeto detalhado.

No projeto de alto nível, o modelo de análise é refinado produzindo uma versão inicial da arquitetura do sistema. Essa atividade é realizada em quatro etapas: a) atribuir papéis aos agentes e definir o diagrama de agentes; b) prover serviços às tarefas, detalhando-as através de um workflow (diagrama de atividades da UML); c) refinar os protocolos de interação através do uso de protocolos de interação da AUML (ODELL; PARUNAK; BAUER, 2000) e de diagramas de estado da UML; e d) especificar o comportamento da interação do papel em diagramas de estado da UML.

O projeto detalhado pressupõe a definição das estruturas de implementação possíveis para o projeto dos agentes, sendo que o MESSAGE define duas abordagens.

A primeira é direcionada por uma arquitetura de agente e por uma organização multiagente, considerando que o agente é uma entidade mais abrangente que uma classe, sendo encarado como um sub-sistema com uma arquitetura interna que define os relacionamentos dos seus diferentes componentes.

A segunda abordagem é mais orientada à plataforma de agente, levando-se em conta que cada

agente pode ser mapeado para uma classe. Este conceito é extraído dos modelos suportados pela maioria das ferramentas para a implementação de agentes, como a JADE (BELLIFEMINE; CAIRE; GREENWOOD, 2007), no qual há uma classe Agente a partir da qual tipos específicos de agentes são derivados.

4.11 INGENIAS

INGENIAS (PAVÓN; GÓMEZ-SANZ; FUENTES, 2005) é uma metodologia que define um conjunto de atividades para modelagem de sistemas multiagente e possui um conjunto de ferramentas de suporte ao desenvolvimento das atividades propostas. Estas ferramentas, assim como a notação usada na metodologia, são baseadas em cinco meta-modelos, que definem as diferentes visões e conceitos pelos quais um sistema multiagente pode ser descrito.

Os conceitos em INGENIAS são descritos através de cinco pontos de vista: organização, agente, objetivos/tarefas, interações e ambiente. Do ponto de vista da organização, é descrito o ambiente onde o sistema será inserido. Tal organização é definida por sua estrutura, funcionalidades e relacionamentos sociais. A estrutura organizacional é definida por grupos e *workflows*. Grupos podem ser definidos como uma sub-organização que possui objetivos e tarefas. A definição das tarefas e quem irá executá-las é definida pelos workflows. O ponto de vista do agente está relacionado a quais objetivos o agente está relacionado, às responsabilidades (que tarefas ele deve executar) e à capacidade (quais papéis ele executa). O ponto de vista de objetivos/tarefas descreve as consequências da execução de uma tarefa tendo em vista o cumprimento de um objetivo, determinando para cada tarefa o que é esperado como saída. O ponto de vista de interação mostra a interação entre os diversos agentes ou entre um agente de software e uma pessoa. O ponto de vista do ambiente define as entidades com as quais o sistema multiagente interage: recursos, outros agentes e aplicações.

As etapas a serem seguidas no processo de desenvolvimento através do INGENIAS podem ser resumidas em: análise, projeto, implementação, verificação e validação.

A etapa de análise é dividida em: i) análise inicial, onde o foco é a definição de um modelo organizacional que reflita a estrutura do sistema multiagente e identifique seus componentes principais (PAVÓN; GÓMEZ-SANZ; FUENTES, 2005); e ii) a análise final, onde a análise inicial é refinada e os objetivos, metas organizacionais e tarefas são identificados.

A fase de projeto lida com questões tecnológicas relativas ao desenvolvimento. A metodologia INGENIAS dá uma ênfase à fase de implementação. Junto com a metodologia foi criado um ambiente de desenvolvimento que oferece suporte à modelagem e geração de código através dos modelos. Este ambiente é chamado de *INGENIAS Development Kit* (IDK).

A etapa de validação e verificação é feita através de módulos específicos do IDK, usando uma técnica conhecida como *Activity Theory* (AT) (FUENTES; GOMEZ-SANZ; PAVON, 2003), que permite estudar as sociedades humanas de forma individual e coletiva.

4.12 MaSE

MaSE (*Multiagent Systems Engineering Methodology*) (DELOACH; KUMAR, 2005) é uma metodologia que define cada agente como uma especialização de um objeto. Assim, MaSE é baseada nas técnicas do paradigma orientado a objetos, aplicando isto no desenvolvimento de sistemas multiagente. Muitos modelos em MaSE são similares à UML, contudo a metodologia possui características específicas aos sistemas multiagente. Para auxílio no processo de construção de sistemas multiagente, na metodologia MaSE é usada uma ferramenta chamada *agentTool* (DELOACH; ENGINEERING; MANAGEMENT, 2001).

A metodologia MaSE pode ser dividida em duas fases principais: Análise e Projeto. Apesar desta divisão, a metodologia tem um caráter iterativo, assim o desenvolvedor tem liberdade para alternar entre as fases sem seguir uma sequência rígida.

A fase de análise tem por objetivo definir um conjunto de modelos que possam ser usados para alcançar os objetivos globais do sistema. Tal fase é dividida em três passos: levantamento de objetivos, especificação de casos de uso e refinamento de papéis.

O primeiro passo, levantamento de objetivos, visa capturar os objetivos do sistema, extraíndo-os dos requisitos pré-existentes, e depois estruturar os objetivos em uma árvore de objetivos. Esta última resulta em um Diagrama de Hierarquia de Objetivos, representando objetivos e relacionamentos entre objetivos e sub-objetivos.

No segundo passo, os objetivos são traduzidos em casos de uso, que capturam os cenários previamente identificados com uma descrição detalhada e um conjunto de diagramas de sequência. Estes casos de uso representam os comportamentos desejados do sistema e a sequência de eventos.

No último passo da fase de análise é feito o refinamento do conjunto inicial de papéis definidos no passo anterior, descrevendo os papéis no sistema e a comunicação entre eles. Cada papel é decomposto em um conjunto de tarefas, que são projetadas para atingir os objetivos de responsabilidade de cada papel. Estas tarefas são documentadas usando um diagrama de tarefas concorrentes. Tarefas concorrentes consistem de um conjunto de estados e transições que representam a comunicação e o raciocínio interno do agente.

A fase de projeto tem por objetivo converter papéis e tarefas em uma forma mais sugestiva para implementação, especialmente os agentes e os diálogos. Tal fase é composta por quatro passos: criação

das classes de agentes, construção dos diálogos, montagem das classes de agentes e projeto do sistema.

O primeiro passo dessa fase é a criação de classes de agentes, que tem o objetivo de identificar as classes de agentes e seus diálogos e, então, documentá-los em diagramas de classes de agentes. O diagrama de classes de agentes, que resulta deste passo, é semelhante ao diagrama de classes orientado a objetos, porém com duas diferenças: as classes de agentes são definidas por papéis ao invés de atributos e os métodos e os relacionamentos entre as classes de agentes são diálogos.

No segundo passo, a construção de diálogos cria um projeto de diálogo detalhado. Diálogos modelam comunicação entre duas classes de agentes usando um par de autômatos de estado finito semelhante, na forma e na função, a tarefas concorrentes. Cada tarefa gera múltiplos diálogos, quando solicita comunicação com mais de uma classe de agente.

O passo seguinte envolve a definição da arquitetura interna de cada agente. MaSE não assume nenhuma arquitetura de agente em particular e permite que seja usada uma grande variedade de arquiteturas já existentes ou propostas novas arquiteturas. A arquitetura do sistema é definida usando componentes semelhantes aqueles definidos na UML.

O passo final da metodologia MaSE implica na definição do sistema a ser implementado, pois já foram definidos os diagramas para mostrar o número, os tipos e as localizações dos agentes no sistema. Nesta etapa, o framework de comunicação selecionado e a linguagem de programação escolhida são utilizados para o processo de desenvolvimento do sistema.

4.13 Estudo Comparativo

Este estudo tem como objetivo a escolha de uma metodologia que seja adequada na realização dos objetivos de desenvolvimento de um sistema multiagente para um ambiente virtual de aprendizagem. Para tal foram considerados alguns parâmetros de comparação. Tais parâmetros se encontram listados na Tabela 4.1 juntamente com o significado das notas dadas.

Tabela 4.1: Parâmetros de comparação das metodologias

Parâmetros	Descrição	Nota 3	Nota 2	Nota 1	Nota 0
Compleitude (Comp.)	Fases do ciclo de vida do software contempladas pela metodologia	Contempla todas as fases	Contempla a maioria das fases	Contempla algumas fases	Não contempla nenhuma fase
Documentação (Doc.)	Documentação da metodologia	Ampla documentação	Possui documentação razoável	Pouca documentação	Não possui documentação
Consolidação (Cons.)	Aceitação pela comunidade de usuários	Grande aceitação e consolidação	Aceitação razoável	Pouca aceitação	Nenhuma aceitação
Adaptação ao contexto (Adapt.)	Adaptação ao contexto da aprendizagem colaborativa	Ampla adaptação	Se adapta com algumas modificações	Se adapta com grandes modificações	É totalmente diferente deste contexto
Ferramentas (Ferram.)	Ferramentas de auxílio às fases da metodologia	Ferramenta para todas as fases	Ferramentas para algumas fases	Ferramenta para uma fase	Nenhuma ferramenta
Aplicabilidade (Apl.)	Aplicabilidade na prática	Muito aplicável na prática	Aplicável razoavelmente	Pouco aplicável	Nenhum uso prático

A Tabela 4.2 mostra as notas dadas para cada metodologia estudada, levando em consideração os parâmetros escolhidos.

Tabela 4.2: Estudo comparativo de metodologias para modelagem de sistemas multiagente

Metodologia	Comp.	Doc.	Cons.	Adapt.	Ferram.	Apl.
MAS-CommonKADS	3	3	3	1	0	1
K-Org	2	1	1	0	0	3
Tropos	1	2	2	2	0	2
PASSI	3	2	3	2	3	3
Prometheus	3	3	2	2	3	3
Gaia	1	2	1	0	0	2
ADELFE	3	2	1	0	3	3
MESSAGE	3	3	1	1	0	3
INGENIAS	3	3	2	1	3	3
MaSE	2	2	2	2	3	3

4.14 Justificativa para escolha da metodologia

Neste estudo foram comparadas dez metodologias com o objetivo de escolher uma que seja adequada na realização dos objetivos de desenvolvimento de um Sistema Multiagente para um ambiente virtual de aprendizagem. Foi constatado que as várias metodologias possuem aplicabilidades diferentes e contextos de aplicações únicas, sendo umas de propósito específico e outras de propósito geral. Concluiu-se também que não é trivial a escolha da melhor metodologia. Para a aplicação em aprendizagem colaborativa destacam-se as seguintes metodologias: MAS-CommonKADS, PASSI, Prometheus e MaSE, pois no estudo comparativo foram as que tiveram melhores pontuações. Contudo, considerando-se os parâmetros adotados, um torna-se eliminatório que é o de Adaptação ao Contexto. Este foi, por exemplo, o motivo pelo qual a metodologia ADELFE foi descartada. Esta metodologia se saiu muito bem nas pontuações, mas destina-se a propósito específico.

A metodologia Prometheus possui um grande diferencial em relação as demais metodologias, uma vez que é bastante completa e detalhada, possuindo ampla documentação e clareza entre as fases. Além de oferecer um passo a passo em direção a solução do problema, ela possui uma ferramenta de modelagem que auxilia no processo. A natureza do problema enfocado neste trabalho também se encaixa no modelo de agentes orientado a metas desta metodologia. Sendo assim, concluímos que a metodologia viável para este trabalho seria a metodologia Prometheus.

Capítulo 5

Arquitetura de Agentes para Suporte à Colaboração na PBL em Ambientes Virtuais de Aprendizagem

Foi apresentado no Capítulo 2 na seção 2.3 diversos sistemas multiagente de apoio à aprendizagem, bem como no Capítulo 3 na subseção 3.2.10 foram apresentados alguns sistemas multiagente que apoiaram uma teoria de aprendizagem em particular. Este capítulo apresenta a arquitetura proposta neste trabalho, como diferencial das demais abordagens referenciadas anteriormente, podemos destacar o fato de nossa arquitetura usar um agente animado de interface que apresenta características sócio-afetivas, ou seja, uma vez que o agente detecta um problema, o agente animado de interface tenta resolver ou minimizar o problema motivando o aluno a participar das atividades e discussões. Para isto, ele se utiliza de expressões faciais, gestos, sons e mensagens textuais, dependendo do grau de passividade detectado. Além disso, diferentemente dos outros trabalhos discutidos, a abordagem proposta tem como base a PBL, que é uma teoria de aprendizagem comprovadamente eficaz (TSENG; CHIANG; HSU, 2008) (STROBEL; BARNEVELD, 2009) (ŞENDAĞ; ODABAŞI, 2009).

Foi apresentado, no capítulo anterior, um estudo comparativo com diversas metodologias de modelagem de sistemas multiagente. Este estudo teve por objetivo motivar a escolha da melhor metodologia para modelagem dos agentes que comporão a arquitetura. Ao final foi justificada a escolha da metodologia Prometheus por diversos motivos apresentados anteriormente. Este capítulo apresentará a modelagem da arquitetura proposta, segundo os modelos e características da metodologia Prometheus. A modelagem foi efetuada com o auxílio da ferramenta Prometheus Design Tool - PDT (PADGHAM; THANGARAJAH; WINIKOFF, 2005). A seção 5.1 apresenta um esquema das diversas fases da metodologia Prometheus. As seções 5.2 e 5.3 apresentam as fases de Especificação de Sistema e Projeto de Arquitetura modeladas para a arquitetura proposta neste trabalho.

5.1 Fases da metodologia Prometheus

A Figura 5.1 apresenta um esquema com as fases da metodologia Prometheus. Conforme podemos ver na Figura 5.1, as fases da metodologia Prometheus se dividem em: Especificação de Sistema, Projeto de Arquitetura e Projeto Detalhado.

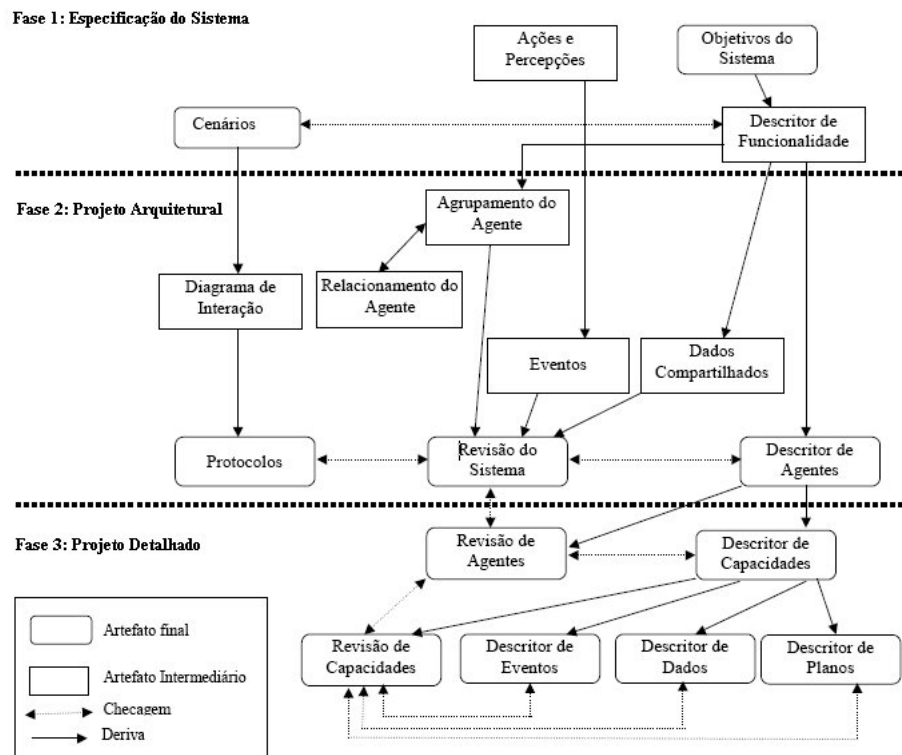


Figura 5.1: Fases da metodologia Prometheus adaptado de (PADGHAM; WINIKOFF, 2002)

5.2 Especificação de Sistema

A Especificação de Sistema consiste na primeira etapa do processo de modelagem do sistema através da metodologia Prometheus. Esta fase tem por objetivo (PADGHAM; WINIKOFF, 2004):

1. Especificar as metas do sistema, resultando numa lista de metas e sub-metas, com descritores associados;
2. Desenvolver um conjunto de cenários, que cobre as metas do sistema e mostra a forma de execução de alguns processos no sistema;

3. Definir um conjunto de funcionalidades que são associadas a uma ou mais metas do sistema e capturam uma parte do comportamento do sistema;
4. Descrever a interface entre os agentes e o ambiente no qual eles estão situados, em termos de percepções e ações.

Ao término desta fase, alguns artefatos são gerados: Diagrama de Metas, Descritores de Metas, Cenários, Diagrama de Funcionalidades e Descritores de Funcionalidades.

5.2.1 Especificação de Metas

A especificação de metas é responsável por capturar as metas do sistema. Esta fase inicial é muito importante, pois fornecerá a base para a construção da arquitetura dos agentes. Uma forma natural de encontrar as principais metas do sistema em uma análise inicial é realizando uma descrição em linguagem natural do funcionamento global do sistema. Abaixo segue essa descrição concernente à arquitetura proposta neste trabalho:

O sistema será desenvolvido seguindo a teoria de aprendizagem baseada em problemas (*Problem-Based Learning - PBL*). Inicialmente o facilitador define o cenário do problema. Este precisa ser definido segundo os requisitos da PBL, ou seja, ele deve ser complexo, mal-estruturado, multidisciplinar, não deve possuir uma única solução e deve exigir reflexão. O sistema deve auxiliar o facilitador na realização desta definição. O grupo de estudantes responsável por este problema em particular deve seguir passos pré-determinados para resolução do problema, seguindo o ciclo de desenvolvimento da PBL. O facilitador é o responsável por guiar os estudantes neste ciclo. O sistema deve auxiliar o facilitador a guiar os estudantes neste processo. O facilitador deve detectar possíveis problemas que possam ocorrer ao longo do processo. O sistema deve auxiliar o facilitador na detecção destes problemas, além de oferecer diretrizes para o uso da melhor estratégia pedagógica afim de corrigir um dado problema. O sistema deve auxiliar o facilitador na formação de grupos afins, através da criação de modelos para cada estudante. Ao final, o sistema deve auxiliar o facilitador na avaliação dos estudantes segundo os critérios estabelecidos na PBL.

Analisando esta breve descrição do sistema é possível extrair um conjunto de metas relevantes:

1. **Cumprimento do ciclo de desenvolvimento da PBL** - Objetiva cumprir o ciclo de desenvolvimento da PBL.
 - (a) **Monitoramento do cumprimento das etapas** - Verificar se as etapas do processo estão sendo cumpridas.

- i. **Auxílio na definição do cenário do problema** - Auxiliar na definição do problema.
 - ii. **Auxílio na análise do problema** - Auxiliar na etapa de análise do problema por parte do grupo de estudantes.
 - A. **Identificação dos fatos** - Identificação de fatos relacionados ao problema.
 - B. **Formulação de hipóteses** - Formulação de hipóteses com base nas questões levantadas. Esta fase é efetuada com conhecimentos prévios dos estudantes.
 - iii. **Avaliação segundo a PBL** - Avaliação do processo, dos estudantes individualmente e do grupo.
 - iv. **Auxílio na resolução do problema** - Resolução efetiva do problema, envolvendo pesquisa e auto-aprendizado.
- (b) **Deteção de problemas** - Detectar problemas no processo de aprendizagem.
- i. **Deteção de problemas de colaboração** - Detectar problemas relacionadas à colaboração dos estudantes, ou seja, verificar se os estudantes estão efetivamente trabalhando em equipe.
 - ii. **Deteção de fuga do contexto** - Detectar problemas relacionadas à fuga do escopo do problema proposto. Isto pode levar ao fracasso da resolução do problema.
 - iii. **Deteção de estagnação do processo** - Detectar a estagnação do processo, que ocorre quando os estudantes não possuem conhecimentos sobre como continuar o processo, ou seja, o grupo sente-se “perdido”.
 - iv. **Deteção de estudantes passivos** - Detectar estudantes passivos, ou seja, aqueles que tem dificuldade de participar de discussões e usar ferramentas de colaboração. Estudantes assim podem entravar o processo e gerar problemas maiores.
 - A. **Diretivas para uso de estratégias pedagógicas** - Oferecer diretivas para uso de estratégias pedagógicas na ocorrência de um problema.
 - B. **Motivação dos estudantes** - Motivar os estudantes a participarem mais das discussões e a usarem as ferramentas colaborativas.
2. **Estímulo à colaboração** - Usar estratégias que aumentem a colaboração e o trabalho em equipe.
- (a) **Criação de modelos de estudantes** - Criar um modelo que reflita o perfil do estudante.
- i. **Monitoração do uso do sistema** - Realizar o monitoramento do uso de ferramentas e execução de atividades por parte do estudante.
- (b) **Recomendação de pares** - Recomendação de pares para auxílio no processo de estudo e resolução do problema.
- (c) **Estabelecimento de metas do grupo** - Estabelecimento de metas do grupo por parte do grupo de estudantes.

- (d) **Gerência de prazos do grupo** - Estabelecimento de prazos e cronograma de atividades.
- (e) **Divisão de responsabilidades** - Divisão de tarefas entre os estudantes concernentes ao problema.
- (f) **Formação de grupos afins** - Formação de grupos considerando afinidade entre os membros para resolução do problema.
 - i. **Comparação de modelos afins** - Comparação de modelos de estudantes com vistas à formação de grupos.
- (g) **Criação de modelos de grupos** - Criação de modelo que reflita o perfil do grupo.

Este conjunto de metas é melhor representado através do diagrama de metas, que é mostrado na Figura 5.2. Apesar de todas estas metas terem sido definidas, nem todas serão contempladas pelos agentes do sistema. Isto se deve ao fato de algumas metas possuírem certas complexidades que as tornam inviáveis a serem cumpridas por um agente de software. Este é o caso das metas: Avaliação segundo a PBL, Auxílio na resolução do problema, Estabelecimento de metas do grupo, Gerência de prazos do grupo e Divisão de responsabilidades.

A meta Avaliação segundo a PBL não foi contemplada por um agente de software pois o processo de avaliação da PBL é muito subjetivo. O facilitador necessita avaliar o estudante e verificar não apenas se ele resolveu o problema, mas também se ele adquiriu as habilidades necessárias ao cumprimento das metas da PBL. Portanto esta meta é atribuída ao facilitador e não a um agente de software.

A meta Auxílio na resolução do problema não foi contemplada por um agente de software, pois o processo de resolução do problema deve ser dirigido pelo grupo de estudantes, o facilitador por outro lado deve guiá-los nesse processo os auxiliando nas diversas etapas, portanto esta meta deve ser atribuída ao facilitador.

As metas Estabelecimento de metas do grupo, Gerência de prazos do grupo e Divisão de responsabilidades não foram contempladas por um agente de software, pois são tarefas que devem ser realizadas pelo líder do grupo, que é o responsável por monitorar o progresso do grupo tendo a responsabilidade de gerência do grupo.

Na próxima etapa apenas as metas que serão associadas a agentes foram relacionadas a alguma funcionalidade.

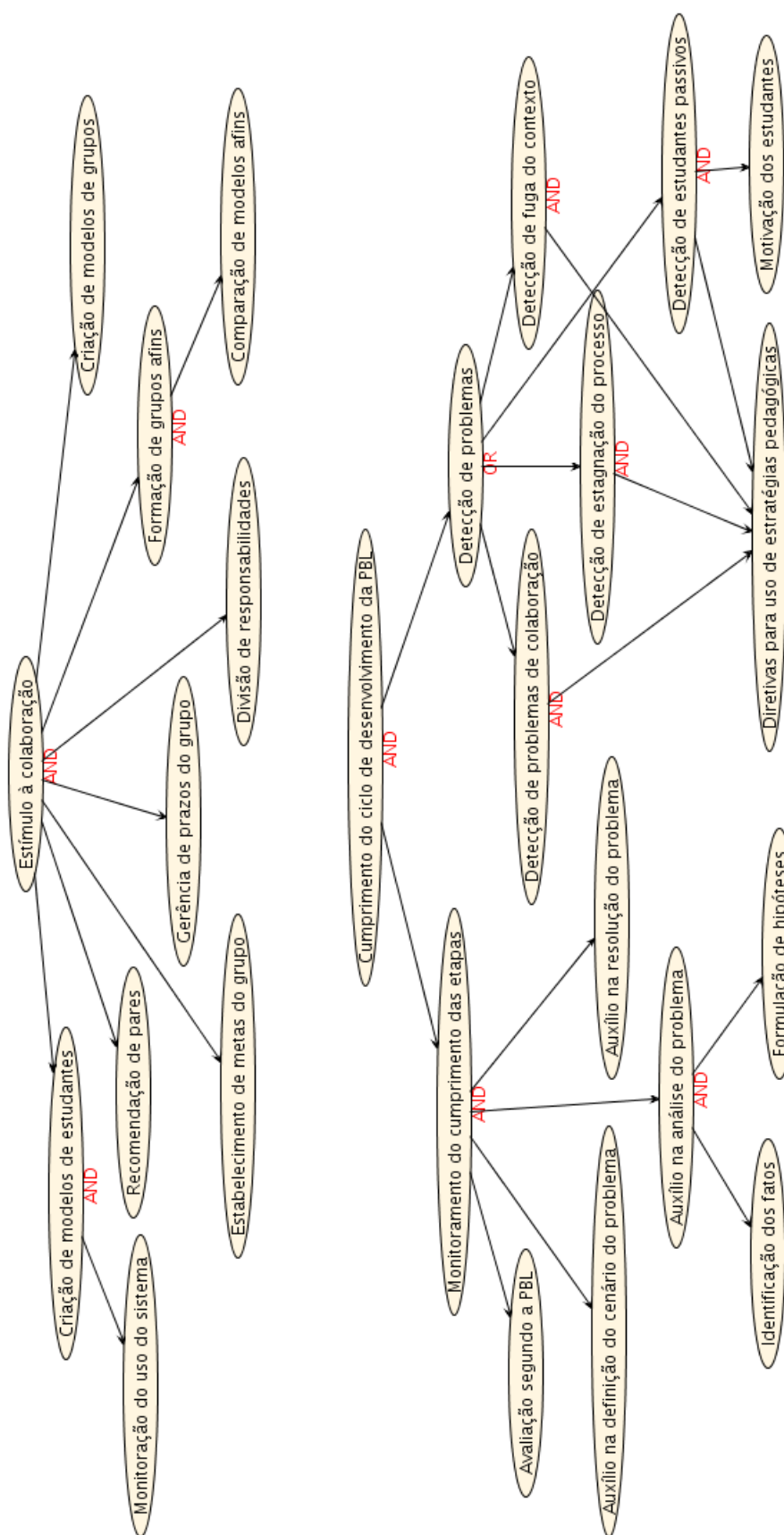


Figura 5.2: Diagrama de Metas

5.2.2 Funcionalidades

Funcionalidade, na metodologia Prometheus, representa um componente que desempenha uma parte do comportamento do sistema. Uma funcionalidade agrupa uma ou mais metas do sistema em torno dela. Estas metas devem ser correlatas, ou seja, precisam pertencer a um mesmo grupo. Uma funcionalidade deve representar uma pequena função no sistema e deve ser descrita de forma breve, não englobando várias funções. Se for este o caso, a funcionalidade deve ser quebrada em duas ou mais funcionalidades conforme seja necessário. Nesta etapa da modelagem são gerados o Diagrama de Funcionalidades, os Descritores de Funcionalidades e a interface entre o ambiente e sistema multiagente.

A Figura 5.3 mostra o Diagrama de Funcionalidades.

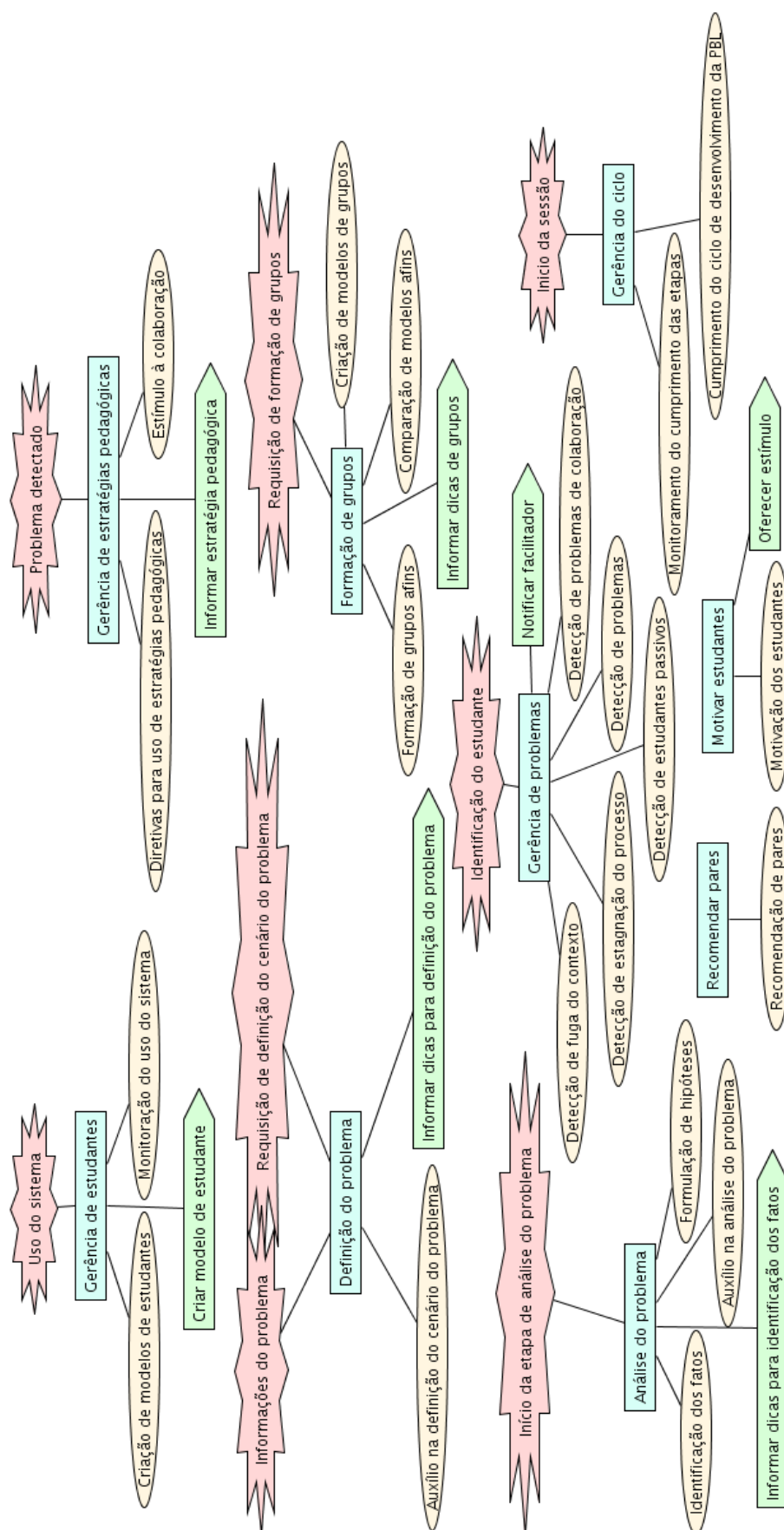


Figura 5.3: Diagrama de Funcionalidades

A legenda para este diagrama é mostrada na Figura 5.4.

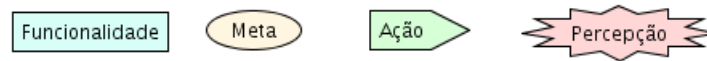


Figura 5.4: Legenda do Diagrama de Funcionalidades

A interface entre o ambiente e o sistema multiagente é descrito em termos de percepções e ações, a seguir são descritas as percepções e ações presentes no Diagrama de Funcionalidades mostrado na Figura 5.3.

Percepções

1. Uso do sistema: percepção relacionada à funcionalidade Gerência de estudantes, representa o uso do sistema pelo estudante que possibilita, posteriormente, a criação de seu modelo;
2. Problema detectado: percepção relacionada à funcionalidade Gerência de estratégias pedagógicas, assim que o problema é detectado é acionado essa funcionalidade;
3. Informações do problema: percepção relacionada à funcionalidade Definição do problema, representa as informações que são passadas pelo usuário no processo definição do problema na PBL;
4. Requisição de definição do problema: percepção relacionada à funcionalidade Definição do problema, representa a requisição efetuada pelo usuário para definição do problema no processo da PBL;
5. Requisição de formação de grupos: percepção relacionada à funcionalidade Formação de grupos, representa a requisição feita por um agente ou usuário do sistema para formação dos grupos;
6. Início da etapa de análise do problema: percepção relacionada à funcionalidade Análise do problema, esta funcionalidade é acionada no início da etapa de análise do problema;
7. Identificação do estudante: percepção relacionada à funcionalidade Gerência de problemas, representa o login do estudante no sistema para verificação de alguns problemas;
8. Início da sessão: percepção relacionada à funcionalidade Gerência do ciclo, representa o início da sessão no processo da PBL.

Ações

1. Criar modelo do estudante: ação relacionada à funcionalidade Gerência de estudantes, representa a ação de criação do modelo do estudante;

2. Informar estratégia pedagógica: ação relacionada à funcionalidade Gerência de estratégias pedagógicas, representa a ação de informar a estratégia pedagógica ao facilitador;
3. Informar dicas para definição do problema: ação relacionada à funcionalidade Definição do problema, representa as dicas oferecidas pelo agente para definição do problema;
4. Informar dicas de grupos: ação relacionada à funcionalidade Formação de grupos, representa as dicas oferecidas pelo agente na formação de grupos;
5. Notificar facilitador: ação relacionada à funcionalidade Gerência de problemas, representa a notificação feita pelo agente ao facilitador na detecção de um problema;
6. Informar dicas para identificação dos fatos: ação relacionada à funcionalidade Análise do problema, representa as dicas oferecidas pelo agente na etapa de identificação de fatos da PBL;
7. Oferecer estímulo: ação relacionada à funcionalidade Motivar estudantes, representa o estímulo dado pelo agente para motivação dos estudantes.

É importante também descrever todas as fontes de dados usadas na arquitetura, a seguir serão descritas todas estas fontes de dados.

Fontes de dados

1. BaseDoProcessoOntologia: Representa a ontologia do ciclo de desenvolvimento da PBL;
2. BaseDeProblemas: É uma base de dados que armazena todos os problemas usados no processo da PBL;
3. BaseEstimulos: É uma base de dados que armazena todos os estímulos usados para motivar os estudantes;
4. BasePerfilDoEstudante: É uma base de dados que armazena o perfil dos estudantes, os campos presentes nesta base são a sua pontuação no uso das ferramentas colaborativas, suas habilidades, deficiências, conhecimentos, faixa etária, sexo e papel;
5. BaseHistoricoEstimulos: É uma base de dados que armazena o histórico dos estímulos usados para motivar os estudantes. Seus campos são o estímulo usado, o estudante estimulado e a data que o estímulo foi usado;
6. BaseOntologiaPalavras: Esta ontologia guarda palavras relacionadas e não relacionadas ao contexto do problema;
7. BaseFerramentasColaborativas: É uma base de dados que guarda as informações sobre o uso das ferramentas colaborativas do ambiente pelos estudantes.

8. BaseDeEstudantes: É uma base que guarda os dados pessoais do estudante, como nome, matrícula e e-mail.
9. PerfilDoGrupoOntologia: Esta ontologia guarda o perfil do grupo;
10. BaseDeGrupo: É uma base de dados que guarda as informações do grupo, como nome do grupo e membros;
11. BaseDeEstrategias: É uma base de dados que guarda as estratégias pedagógicas usadas na correção de problemas que ocorrem no processo de aprendizagem.

Os descritores das funcionalidades presentes no diagrama de funcionalidades são mostrados nas tabelas 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8 e 5.9.

Tabela 5.1: Funcionalidade Definição do Problema

Nome	Funcionalidade Definição do Problema
Descrição	Esta funcionalidade gerencia a definição do problema na PBL. Tem como objetivo informar características relevantes ao problema.
Metas	Auxílio na definição do cenário do problema.
Percepções	Informações do problema e requisição de definição do cenário do problema.
Ações	Informar dicas para definição do problema.

Tabela 5.2: Funcionalidade Análise do Problema

Nome	Funcionalidade Análise do Problema
Descrição	Auxilia na identificação de aspectos relevantes ao problema por parte do grupo de estudantes envolvidos.
Metas	Auxílio na análise do problema; Identificação dos fatos; Formulação de hipóteses.
Percepções	Início da etapa de análise do problema.
Ações	Informar dicas para identificação dos fatos.

Tabela 5.3: Funcionalidade Gerência do Ciclo

Nome	Funcionalidade Gerência do Ciclo
Descrição	Esta funcionalidade gerencia o ciclo de desenvolvimento da PBL. Ela garante que as etapas do ciclo de desenvolvimento serão cumpridas.
Metas	Monitoramento do cumprimento das etapas; Cumprimento do ciclo de desenvolvimento da PBL.
Percepções	Início da sessão.
Ações	-

Tabela 5.4: Funcionalidade Gerência de Problemas

Nome	Funcionalidade Gerência de Problemas
Descrição	Esta funcionalidade é responsável por realizar o gerenciamento da detecção de problemas que possivelmente ocorrem no processo tutorial.
Metas	Detecção de Problemas; Detecção de fuga do contexto; Detecção de estagnação do processo; Detecção de estudantes passivos; Detecção de problemas de colaboração.
Percepções	Identificação do Estudante.
Ações	Notificar facilitador.

Tabela 5.5: Funcionalidade Gerência de Estratégias Pedagógicas

Nome	Funcionalidade Gerência de Estratégias Pedagógicas
Descrição	Esta funcionalidade tem por objetivo gerenciar a geração de diretivas para a definição da estratégia pedagógica a ser usada na presença de um problema. Ela ajudará o facilitador a tomar uma posição dado o surgimento de um problema no processo da PBL.
Metas	Estímulo à colaboração; Diretivas para uso de estratégias pedagógicas.
Percepções	Problema detectado
Ações	Informar estratégia pedagógica.

Tabela 5.6: Funcionalidade Gerência de Estudantes

Nome	Funcionalidade Gerência de Estudantes
Descrição	Esta funcionalidade tem por objetivo gerenciar as atividades executadas pelo estudante, sendo responsável por realizar o monitoramento das atividades e criação de perfis de estudantes.
Metas	Monitoração do uso do sistema; Criação de modelos de estudantes.
Percepções	Uso do sistema
Ações	Criar modelo de estudante.

Tabela 5.7: Funcionalidade Formação de Grupos

Nome	Funcionalidade Formação de Grupos
Descrição	Esta funcionalidade tem por objetivo realizar o gerenciamento da formação de grupos.
Metas	Comparação de modelos afins; Formação de grupos afins; Criação de modelos de grupos.
Percepções	Requisição de formação de grupos.
Ações	Informar dicas de grupos.

Tabela 5.8: Funcionalidade Recomendar Pares

Nome	Funcionalidade Recomendar Pares
Descrição	Esta funcionalidade tem por objetivo realizar o processo de recomendação de pares de estudantes.
Metas	Recomendação de pares.
Percepções	-
Ações	-

Tabela 5.9: Funcionalidade Motivar Estudantes

Nome	Funcionalidade Motivar Estudantes
Descrição	Esta funcionalidade tem por objetivo realizar o processo de motivação dos estudantes.
Metas	Motivação dos estudantes.
Percepções	-
Ações	Oferecer estímulo.

5.2.3 Desenvolvimento de Cenários

Os cenários mostram uma sequência de passos que descrevem o fluxo do processo no sistema para atingir uma dada meta. Em outras palavras, tentam demonstrar um processo comum sendo executado no sistema. Em alguns casos, na etapa de desenvolvimento de cenários, outras metas são desenvolvidas para complementar aspectos que antes não foram visualizados em uma análise prévia. A Figura 5.5 mostra o diagrama de cenários, contendo um conjunto de cenários e a relação entre eles. Em seguida é descrito cada cenário individualmente.

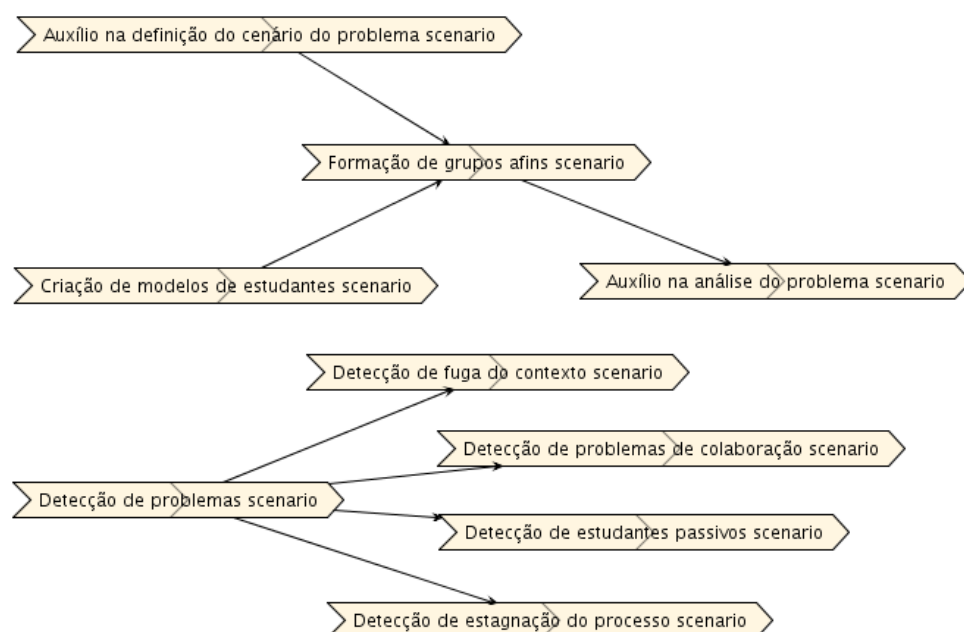


Figura 5.5: Diagrama de Cenários

Tabela 5.10: Cenário Auxílio na Definição do Cenário do Problema

Nome	Cenário Auxílio na Definição do Cenário do Problema						
Descrição	Cenário que descreve a etapa de definição do problema feita pelo facilitador.						
Trigger	Início da sessão PBL						
Passos	#	Tipo	Nome	Funcionalidade associada	Descrição do passo	Dados usados	Dados produzidos
	1	Percepção	Requisição de definição do cenário do problema	Definição do problema	Requisição feita pelo facilitador para definição do problema	-	-
	2	Meta	Cumprimento do ciclo de desenvolvimento da PBL	Gerência do ciclo	Cumprimento do ciclo de desenvolvimento da PBL	-	-
	3	Meta	Monitoramento do cumprimento das etapas	Gerência do ciclo	Monitoramento para verificar se as etapas do processo estão sendo cumpridas	-	-
	4	Percepção	Informações do problema	Definição do problema	Receber informações relativas ao problema	-	BaseDeProblemas
	5	Ação	Informar dicas para definição do problema	Definição do problema	O sistema oferece dicas para definição do problema	BaseDeProblemas	BaseDeProblemas
	6	Cenário	Cenário formação de grupos afins	-	-	-	-

Tabela 5.11: Cenário Criação de Modelos de Estudantes

Nome	Cenário Criação de Modelos de Estudantes						
Descrição	Cenário que descreve a criação de modelos de estudantes.						
Trigger	Realizado por agente devido ao monitoramento do uso do sistema pelos estudantes.						
Passos	#	Tipo	Nome	Funcionalidade associada	Descrição do passo	Dados usados	Dados produzidos
	1	Meta	Monitoração do uso do sistema	Gerência de estudantes	Monitoramento do uso do sistema por parte do estudante	BaseDeEstudantes, BaseFerramentasColaborativas	-
	2	Ação	Criar modelo do estudante	Gerência de estudantes	Criação do modelo do estudante com base no monitoramento do seu uso e seu perfil	-	BasePerfilDoEstudante
	3	Cenário	Cenário Formação de grupos afins	-	-	-	-

Tabela 5.12: Cenário Formação de Grupos Afins

Nome	Cenário Formação de Grupos Afins						
Descrição	Cenário que descreve o processo de formação de grupos.						
Trigger	Requisição do facilitador no início ou durante o processo. Feito de forma autônoma por um agente.						
Passos	#	Tipo	Nome	Funcionalidade associada	Descrição do passo	Dados usados	Dados produzidos
	1	Percepção	Requisição de formação de grupos	Formação de grupos	Requisição de formação de grupos pelo facilitador	-	-
	2	Meta	Comparação de modelos afins	Formação de grupos	Comparação de modelos de estudantes	BaseDeEstudantes	-
	3	Ação	Informar dicas de grupos	Formação de grupos	Informar ao facilitador possíveis formações de grupos	BaseDeEstudantes, BaseDeGrupos	-
	4	Cenário	Cenário auxílio na análise do problema	-	-	-	-

Tabela 5.13: Cenário Auxílio na Análise do Problema

Nome	Cenário Auxílio na Análise do Problema						
Descrição	Este cenário descreve o processo de auxílio na análise do problema por parte do grupo de estudantes.						
Trigger	Início da etapa de análise do problema, que ocorre posteriormente à formação de grupos e à etapa de definição do cenário do problema.						
Passos	#	Tipo	Nome	Funcionalidade associada	Descrição do passo	Dados usados	Dados produzidos
	1	Percepção	Início da etapa de análise do problema	Análise do problema	O agente recebe a percepção que iniciou esta etapa	-	-
	2	Meta	Identificação dos fatos	Análise do problema	Auxílio na identificação dos fatos relacionados ao problema em questão	BaseDo Processo Ontologia, BaseDeProblemas	-
	3	Ação	Informar dicas para identificação dos fatos	Análise do problema	Dicas para identificação dos fatos	-	-
	4	Outro	Elaboração de questões de aprendizado	-	Elaboração por parte dos estudantes das questões de aprendizado	-	-
	5	Meta	Formulação de hipóteses	Análise do problema	Auxílio na formulação de hipóteses baseado no conhecimento de cada um	BaseDo Processo Ontologia, BaseDeProblemas	-

Tabela 5.14: Cenário Detecção de Fuga do contexto

Nome	Cenário Detecção de Fuga do Contexto						
Descrição	Este cenário descreve o processo de detecção de fuga do contexto do problema.						
Trigger	Agente detecta fuga do contexto						
Passos	#	Tipo	Nome	Funcionalidade associada	Descrição do passo	Dados usados	Dados produzidos
	1	Outro	-	Gerência de problemas	Monitoramento das discussões para detecção de assuntos não relativos ao problema em questão	Base Ferramentas Colaborativas	-
	2	Outro	-	Gerência de problemas	Comparar discussão com ontologia de palavras relacionadas ou não relacionadas	BaseOntologia Palavras	-
	3	Meta	Diretivas para uso de estratégias pedagógicas	Gerência de estratégias pedagógicas	Meta relacionada ao uso da estratégia pedagógica	BaseDe Estrategias	-
	4	Ação	Informar estratégia pedagógica	Gerência de estratégias pedagógicas	O agente oferece dica ao facilitador para uso de estratégia pedagógica baseado no problema de fuga do contexto do problema	-	-
	5	Meta	Motivação do estudante	Motivar estudantes	Após a detecção de fuga do contexto realizar a motivação do estudante	BaseDe Estimulos, BaseHistoricoEstimulos	BaseHistorico Estimulos

Tabela 5.15: Cenário Detecção de Problemas de Colaboração

Nome	Cenário Detecção de Problemas de Colaboração						
Descrição	Este cenário descreve o processo de detecção de problemas de colaboração.						
Trigger	Agente detecta problemas de colaboração						
Passos	#	Tipo	Nome	Funcionalidade associada	Descrição do passo	Dados usados	Dados produzidos
	1	Outro	-	Gerência de problemas	Monitoramento do uso das ferramentas colaborativas; monitoramento de alocação de tarefas; monitoramento do conteúdo das discussões	Base Ferramentas Colaborativas	-
	2	Meta	Conselhos para uso de estratégia pedagógica	Gerência de estratégias pedagógicas	Meta relacionada ao uso da estratégia pedagógica	BaseDe Estrategias	-
	3	Ação	Informar estratégia pedagógica	Gerência de estratégias pedagógicas	O agente oferece dica ao facilitador para uso de estratégia pedagógica baseado no problema de colaboração	-	-

Tabela 5.16: Cenário Detecção de Estudantes Passivos

Nome	Cenário Detecção de Estudantes Passivos						
Descrição	Este cenário descreve o processo de identificação de estudantes passivos.						
Trigger	Agente detecta problema de estudante passivo						
Passos	#	Tipo	Nome	Funcionalidade associada	Descrição do passo	Dados usados	Dados produzidos
	1	Outro	-	Gerência de problemas	Monitoramento das atividades do estudante	Base Ferramentas Colaborativas	BasePerfilDo Estudante
	2	Ação	Notificar facilitador	Gerência de problemas	Notificar o facilitador da ocorrência de estudante passivo	-	-
	3	Meta	Diretivas para uso de estratégias pedagógicas	Gerência de estratégias pedagógicas	Meta relacionada ao uso da estratégia pedagógica	BaseDe Estrategias	-
	4	Ação	Informar estratégia pedagógica	Gerência de estratégias pedagógicas	O agente oferece dica ao facilitador para uso de estratégia pedagógica baseado no problema de estudante passivo	-	-
	5	Meta	Motivação do estudante	Motivar estudantes	Após a detecção de estudantes passivos realizar a motivação do estudante	BaseDe Estimulos, BaseHistoricoEstimulos	BaseHistorico Estimulos

Tabela 5.17: Cenário Detecção de Estagnação do Processo

Nome	Cenário Detecção de Estagnação do Processo						
Descrição	Este cenário descreve o processo de detecção do problema de estagnação no processo.						
Trigger	Agente detecta problema de estagnação do processo						
Passos	#	Tipo	Nome	Funcionalidade associada	Descrição do passo	Dados usados	Dados produzidos
	1	Outro	-	Gerência de problemas	Monitoramento dos resultados para verificar se o grupo está progredindo	-	-
	2	Meta	Diretivas para uso de estratégias pedagógicas	Gerência de estratégias pedagógicas	Meta relacionada ao uso da estratégia pedagógica	BaseDe Estrategias	-
	3	Ação	Informar estratégia pedagógica	Gerência de estratégias pedagógicas	O agente oferece dica ao facilitador para uso de estratégia pedagógica baseado no problema de estagnação do processo	-	-

5.3 Projeto de Arquitetura

Após a conclusão da primeira etapa da modelagem, a especificação de sistema, a próxima etapa é o projeto de arquitetura. Esta etapa é dividida em três atividades (PADGHAM; WINIKOFF, 2004):

1. Decidir os tipos de agentes que comporão o sistema;
2. Descrever as interações entre os agentes no sistema através de diagramas de interação e protocolos de interação;
3. Desenvolver a arquitetura global do sistema, através de um diagrama de sistema panorâmico. Este diagrama captura os tipos de agentes do sistema, além dos limites deste em termos de percepções e ações.

5.3.1 Decidindo os Tipos de Agentes

O passo mais importante da etapa do projeto de arquitetura é a atividade de decidir os tipos de agentes que comporão a arquitetura. Este processo é realizado da seguinte forma:

1. Agrupar funcionalidades em torno dos agentes;

2. Rever o agrupamento considerando o diagrama de conhecimento dos agentes;
3. Desenvolver os descritores de agentes.

5.3.1.1 Agrupando funcionalidades

Uma forma de decidir os tipos de agentes que comporão a arquitetura é agrupando as funcionalidades em torno dos agentes. No sistema existem funcionalidades que são correlatas. Uma das formas de verificar isto é analisando o diagrama de acoplamento de dados, que mostra o relacionamento entre as funcionalidades e as diversas fontes de dados do sistema. A lógica usada aqui é que, se as funcionalidades estão relacionadas a fontes de dados correlatas, elas naturalmente estão relacionadas entre si. O diagrama de acoplamento de dados da arquitetura de agentes proposta neste trabalho é mostrado na Figura 5.6.

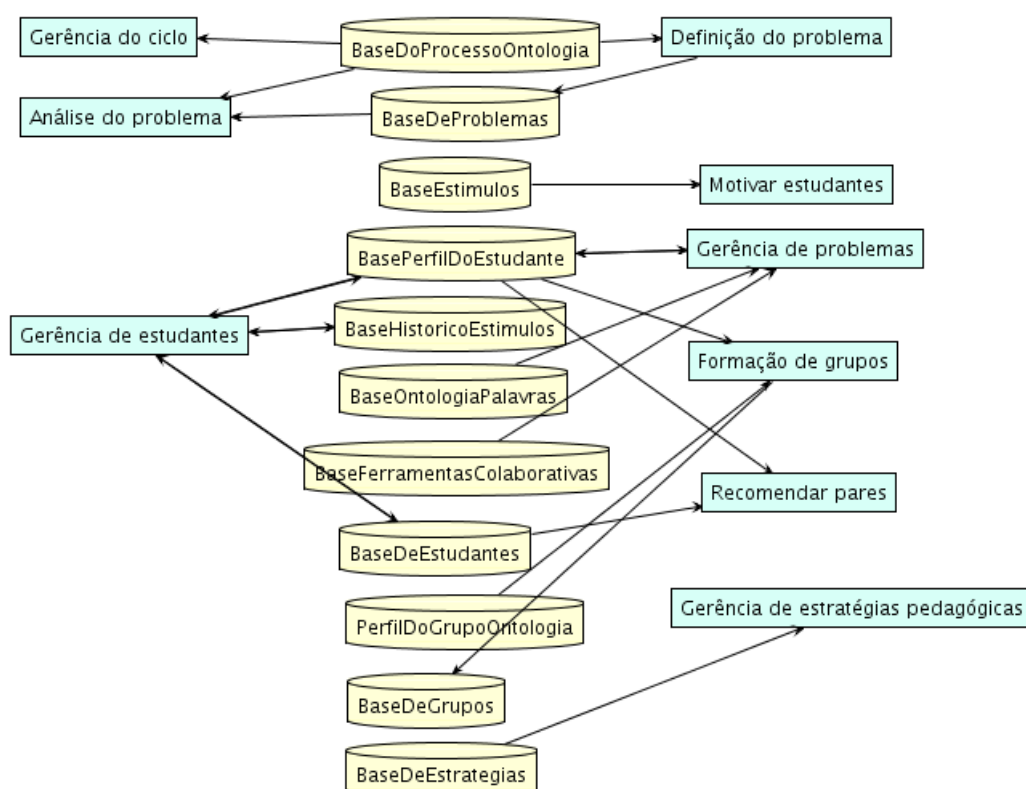


Figura 5.6: Diagrama de Acoplamento de Dados

Analisando o diagrama de acoplamento de dados e as funções inerentes ao sistema, é possível realizar o agrupamento das funcionalidades em torno dos agentes. O diagrama de agente-funcionalidade, que está ilustrado na Figura 5.7, apresenta este agrupamento.

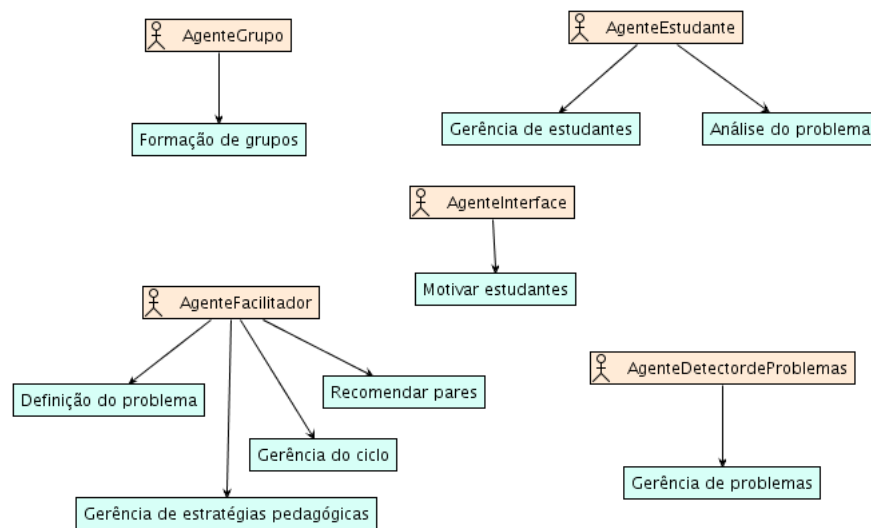


Figura 5.7: Diagrama de Agente-Funcionalidade

5.3.1.2 Revendo agrupamento

O Diagrama de conhecimento de agente ilustrado na Figura 5.8 é uma forma sucinta de verificar as interações entre os diversos agentes do sistema. Neste diagrama podemos ver todos os agentes e o relacionamento entre eles. As setas no diagrama mostram a forma de comunicação, podendo ser ela unidirecional ou bidirecional. Mais à frente será descrita esta interação com mais detalhes através dos protocolos de interação.

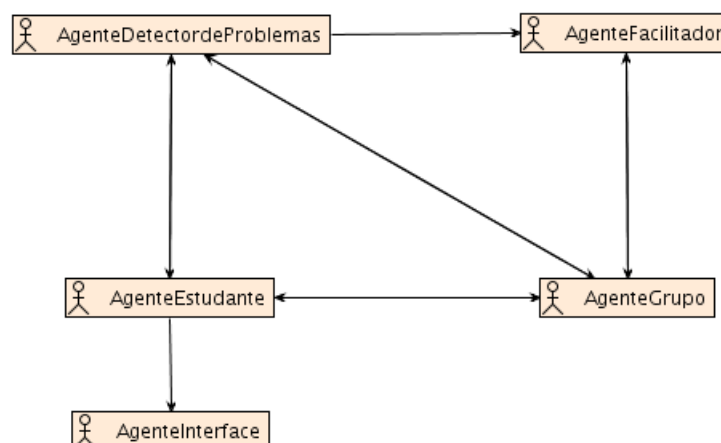


Figura 5.8: Diagrama de Conhecimento do Agente

5.3.1.3 Descrevendo os agentes

Nesta etapa serão mostrados os descritores dos agentes. Estes descritores mostram uma visão geral dos agentes e suas funções, além de todas as entidades as quais ele está relacionado. Os descritores são mostrados nas tabelas 5.18, 5.19, 5.20, 5.21 e 5.22.

Tabela 5.18: Descritor AgenteFacilitador

Nome	AgenteFacilitador
Descrição	Agente responsável por gerir o ciclo de desenvolvimento da PBL; auxiliar na definição do problema; e oferecer dicas para uso de estratégias pedagógicas.
Cardinalidade	um/facilitador
Tempo de vida	Criado no cadastro de facilitador. Termina suas ações na exclusão do facilitador ou sua inatividade.
Funcionalidades incluídas	Gerência de estratégias pedagógicas; Definição do problema, Gerência do ciclo; Recomendar pares.
Usa dados	BaseDoProcessoOntologia, BaseDeProblemas, BaseDeEstrategias, BaseDeEstudantes, BasePerfilEstudante
Produz dados	BaseDeProblemas
Metas	Diretivas para uso de estratégias pedagógicas; Estímulo à colaboração; Auxílio na definição do cenário do problema; Monitoramento do cumprimento das etapas; Recomendação de pares; Cumprimento do ciclo de desenvolvimento da PBL.
Responde percepções	Informações do problema; Requisição de definição do cenário do problema; Início da sessão; Problema detectado.
Executa ações	Informar dicas para definição do problema; Informar estratégia pedagógica.

Tabela 5.19: Descritor AgenteDetectordeProblemas

Nome	AgenteDetectordeProblemas
Descrição	Responsável por detectar problemas no ambiente e informar ao facilitador para que este possa usar a melhor estratégia pedagógica.
Cardinalidade	um/sistema
Tempo de vida	Criado no início do processo e perdura por tempo indeterminado.
Funcionalidades incluídas	Gerência de problemas
Usa dados	BasePerfilDoEstudante, BaseFerramentasColaborativas, BaseOntologiaPalavras
Produz dados	BasePerfilDoEstudante
Metas	Deteção de problemas; Deteção de fuga do contexto; Deteção de estagnação do processo; Deteção de estudantes passivos; Deteção de problemas de colaboração.
Responde percepções	Identificação do estudante
Executa ações	Notificar facilitador

Tabela 5.20: Descritor AgenteEstudante

Nome	AgenteEstudante
Descrição	Responsável pela gerência dos estudantes.
Cardinalidade	um/estudante
Tempo de vida	Criado no cadastro do estudante e finalizado na sua exclusão.
Funcionalidades incluídas	Gerência de estudantes; Análise do problema.
Usa dados	BasePerfilDoEstudante, BaseDeEstudantes, BaseHistoricoEstimulos
Produz dados	BasePerfilDoEstudante, BaseDeEstudantes, BaseHistoricoEstimulos
Metas	Criação de modelos de estudantes; Monitoração do uso do sistema; Identificação dos fatos; Auxílio na análise do problema; Formulação de hipóteses.
Responde percepções	Início da etapa de análise do problema; Uso do sistema.
Executa ações	Criar modelo de estudante; Informar dicas para identificação de fatos.

Tabela 5.21: Descritor AgenteGrupo

Nome	AgenteGrupo
Descrição	Agente responsável pela formação e gerência dos grupos.
Cardinalidade	um/grupo
Tempo de vida	Criado no início e perdura por tempo indeterminado.
Funcionalidades incluídas	Formação de grupos
Usa dados	BaseDeGrupos, BasePerfilDoEstudante, PerfilDoGrupoOntologia
Produce dados	BaseDeGrupos
Metas	Formação de grupos afins; Comparação de modelos afins; Criação de modelos de grupos.
Responde percepções	Requisição de formação de grupos.
Executa ações	Informar dicas de grupos.

Tabela 5.22: Descritor AgenteInterface

Nome	AgenteInterface
Descrição	Agente responsável pela Interface do usuário.
Cardinalidade	um/sistema
Tempo de vida	Criado no início e perdura por tempo indeterminado.
Funcionalidades incluídas	Motivar estudantes
Usa dados	BaseEstimulos
Produce dados	-
Metas	Motivação dos estudantes
Responde percepções	-
Executa ações	Oferecer estímulo

5.3.2 Descrevendo Interações entre os Agentes

Em sistemas multiagente a comunicação é essencial para que o processo colaborativo seja bem sucedido. Na modelagem dos agentes é necessário especificar os protocolos de interação entre os agentes. Nesta modelagem em particular, dois protocolos de interação foram desenvolvidos: o protocolo de detecção de problemas e o protocolo de formação de grupos. Os diagramas foram desenvolvidos usando a linguagem AUMML (ODELL; PARUNAK; BAUER, 2000), que é usada para especificar a interação entre os agentes. A metodologia Prometheus não possui uma forma própria de especificar detalhadamente esta interação, fazendo uso da AUMML.

5.3.2.1 Protocolo de Detecção de Problemas

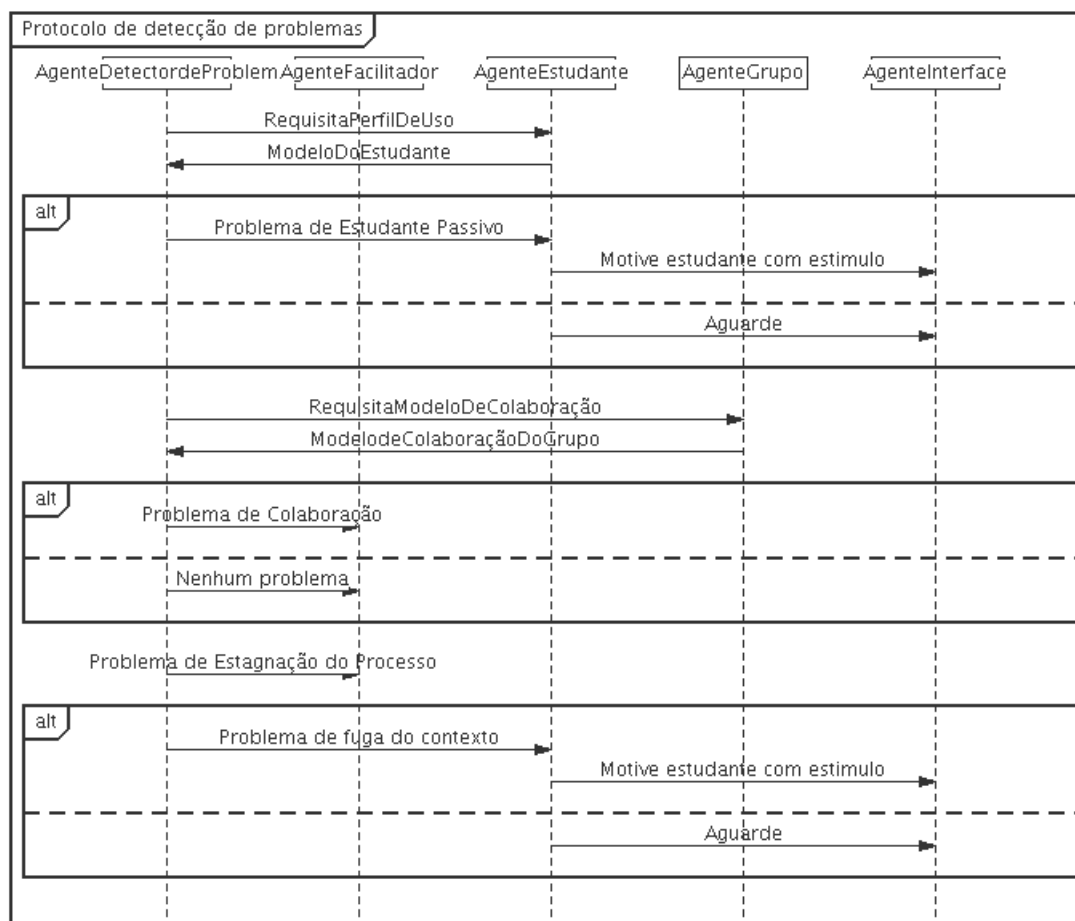


Figura 5.9: Protocolo de detecção de problemas

A Figura 5.9 mostra o protocolo de detecção de problemas. O processo de detecção de problemas envolve todos os agentes e, conforme mostra a figura, ocorre da seguinte forma:

1. Para detecção de estudantes passivos, o AgenteDetectordeProblemas requisita o perfil de uso do sistema ao AgenteEstudante; o AgenteEstudante requisita informações do estudante; o AgenteDetectordeProblemas oferece esta informação; o AgenteEstudante informa o perfil de uso do estudante no sistema e o AgenteDetectordeProblemas, caso detecte um estudante passivo, informa ao AgenteEstudante, que irá acionar o AgenteInterface para motivar o estudante com um estímulo, ou pedir que ele aguarde¹.

¹O AgenteEstudante pode pedir ao AgenteInterface que apenas aguarde e não faça nada até que ele informe o contrário. Isto é importante para evitar interrupções desnecessárias do AgenteInterface, que podem prejudicar o andamento do processo pedagógico.

2. Para detecção de problemas de colaboração, o AgenteDetectordeProblemas requisita o modelo de colaboração ao AgenteGrupo; o AgenteGrupo requisita informações do grupo; o AgenteDetectordeProblemas oferece esta informação; o AgenteGrupo informa o modelo de colaboração do grupo no sistema e o AgenteDetectordeProblemas pode informar ao AgenteFacilitador a ocorrência de um problema de colaboração ou não.
3. Para detecção de problemas de estagnação do processo, o AgenteDetectordeProblemas utiliza outros processos, como monitoramento das discussões e andamento do progresso do grupo.
4. Para detecção de problemas de fuga do contexto, o AgenteDetectordeProblemas irá monitorar o uso das ferramentas colaborativas e, caso detecte este problema, ele irá acionar o AgenteInterface para motivar o estudante com um estímulo, ou pedir que ele aguarde.

5.3.2.2 Protocolo de Formação de Grupos

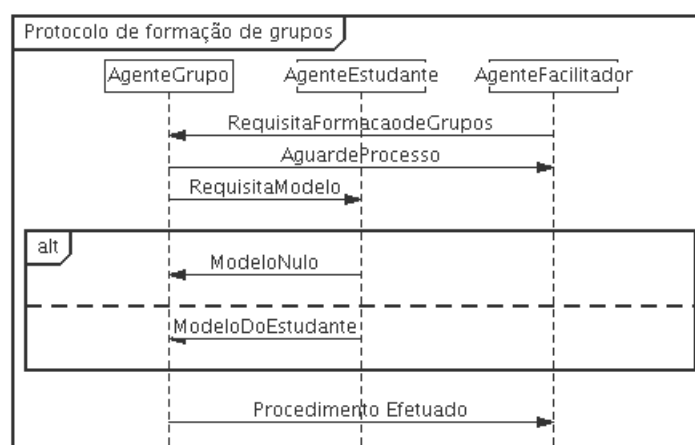


Figura 5.10: Protocolo de formação de grupos

A Figura 5.10 mostra o protocolo de formação de grupos. O processo de formação de grupos envolve o AgenteGrupo, o AgenteEstudante e o AgenteFacilitador e, conforme mostra a figura, ocorre da seguinte forma:

1. O AgenteFacilitador requisita formação de grupos ao AgenteGrupo;
2. O AgenteGrupo envia uma mensagem ao AgenteFacilitador solicitando que aguarde²;

²O AgenteGrupo precisa se comunicar com o AgenteEstudante antes de oferecer uma resposta ao AgenteFacilitador. Por esse motivo precisa enviar uma mensagem ao facilitador para aguardar a conclusão deste processo.

3. O AgenteGrupo requisita ao AgenteEstudante os modelos dos estudantes;
4. O AgenteEstudante requisita informações dos estudantes e o AgenteGrupo oferece esta informação;
5. O AgenteEstudante informa ao AgenteGrupo os perfis dos estudantes;
6. O AgenteGrupo informa ao facilitador a composição dos grupos a serem formados.

Vale salientar que este protocolo de interação descreve o processo inicial de formação de grupos ao longo do uso do sistema. O AgenteGrupo pode requisitar ao AgenteEstudante perfis atualizados e oferecer novas dicas de formação de grupos.

5.3.3 Diagrama da Estrutura Global do Sistema

Após realizar partes da modelagem isoladamente, uma etapa muito importante é o desenvolvimento do diagrama da estrutura global do sistema, que mostra de um forma global as diversas partes que compõem o sistema multiagente. Este diagrama é mostrado na Figura 5.11.

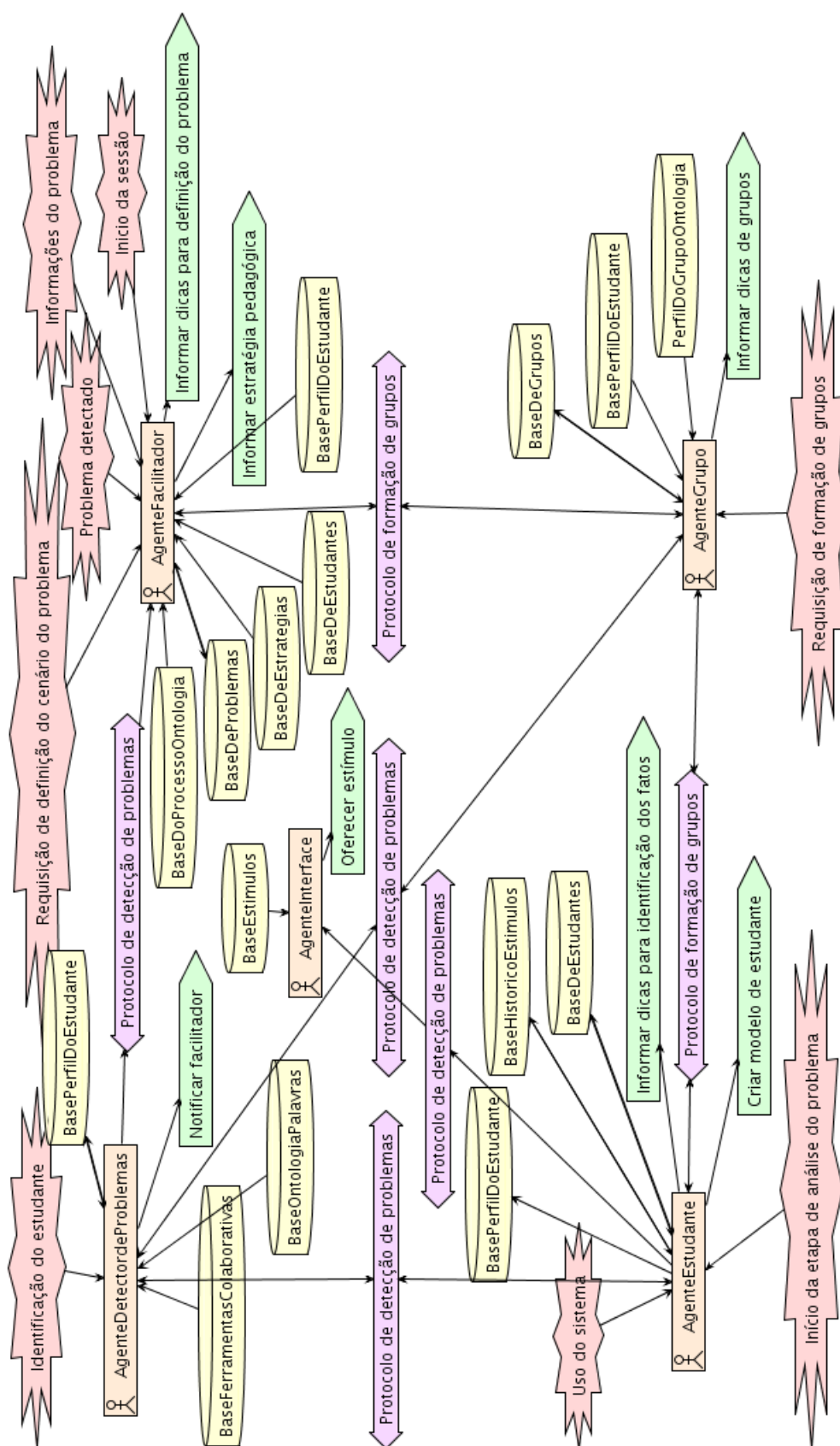


Figura 5.11: Diagrama da estrutura global do sistema

O diagrama da Figura 5.11 apresenta uma visão geral de todos os agentes envolvidos no suporte à colaboração na PBL em ambientes virtuais de aprendizagem. Neste diagrama podemos ver também todas as percepções e ações necessárias a estes agentes para possibilitar que eles cumpram as metas com eficácia, além de todas as fontes de dados que os agentes usam e/ou produzem no processo. Além disso, o diagrama contempla a comunicação entre eles através dos protocolos de interação.

No Capítulo 1, na seção Problemática, foram mostrados alguns problemas referentes à implementação da PBL. Veremos neste momento como a arquitetura, proposta neste trabalho, auxilia na resolução de alguns destes problemas.

O primeiro problema afirma que não é trivial completar o ciclo de desenvolvimento da PBL. Na arquitetura proposta temos a meta Cumprimento do ciclo de desenvolvimento da PBL e suas sub-metas que auxiliam no cumprimento de algumas etapas do ciclo da PBL, facilitando o cumprimento deste ciclo no ambiente. Esta meta está relacionada à funcionalidade Gerência do ciclo que por sua vez é de responsabilidade do agente facilitador.

O segundo problema afirma que a construção do cenário do problema não é fácil, pois os problemas devem ser complexos e mal estruturados, além de permitirem reflexão por parte dos estudantes. A meta Auxílio na definição do cenário do problema auxilia o processo de construção do problema, facilitando este processo.

O terceiro problema afirma que o processo de aprendizado é complexo, pois deve envolver vários objetos de estudos e deve ser multidisciplinar. A meta Auxílio na análise do problema auxilia no processo de aprendizagem, permitindo que os estudantes possam analisar o problema com o auxílio dos agentes. Esta meta está relacionada à funcionalidade Análise do problema que por sua vez é de responsabilidade do agente estudante.

O quarto problema afirma que é difícil para o facilitador garantir uma colaboração efetiva. A meta Estímulo à colaboração e suas sub-metas (Criação de modelos de estudantes, Monitoração do uso do sistema, Recomendação de pares, Formação de grupos afins e Criação de modelos de grupos) auxiliam no estímulo à colaboração, permitindo que o facilitador possa ser auxiliado para garantir uma colaboração efetiva no ambiente. A meta Estímulo à colaboração está relacionada à funcionalidade Gerência de estratégias pedagógicas, que por sua vez é de responsabilidade do agente facilitador.

O quinto problema afirma que é difícil para o facilitador monitorar o processo de aprendizagem, pois é auto-dirigido pelo estudante de forma que o facilitador não pode oferecer as respostas, apenas guiá-lo neste processo. A meta Monitoramento do cumprimento das etapas auxilia no monitoramento do processo, oferecendo subsídio ao facilitador para que ele possa realizar de forma efetiva o monitoramento do processo. Esta meta está relacionada à funcionalidade Gerência do ciclo que por sua vez é de responsabilidade do agente facilitador.

Ao final, é dito que a implantação da PBL na EaD é ainda mais complexa, pois o facilitador não está

presente fisicamente nas seções e nem sempre pode detectar possíveis problemas que possam ocorrer no processo de aprendizagem. Para isso, a arquitetura possui a meta Detecção de problemas e suas sub-metas: Detecção de problemas de colaboração, Detecção de estagnação do processo, Detecção de fuga do contexto, Detecção de estudantes passivos, Diretivas para uso de estratégias pedagógicas e Motivação dos estudantes. Estas metas auxiliam no processo de detecção e auxílio na correção de possíveis problemas que possam ocorrer no processo de aprendizagem. A meta Detecção de problemas está relacionada à funcionalidade Gerência de problemas, que por sua vez é de responsabilidade do agente detector de problemas. Desta forma, concluímos que a arquitetura apresentada possibilita o uso efetivo da PBL em ambientes virtuais de aprendizagem.

Neste capítulo foi feita a modelagem da arquitetura até a etapa do projeto de arquitetura da metodologia Prometheus. O projeto detalhado (interior) de alguns agentes será realizado no próximo capítulo.

Capítulo 6

Estudo de Caso: Detecção de Estudantes Passivos e Fuga de Contexto na PBL

No Capítulo 5 foi visto a modelagem da arquitetura de agentes para suporte à colaboração na PBL em ambientes virtuais de aprendizagem com a metodologia Prometheus. Neste capítulo será apresentado o estudo de caso com a implementação dos agentes necessários para o alcance de duas metas da arquitetura proposta: detecção de estudantes passivos e detecção de conversações fora de contexto do problema sendo trabalhado na PBL no ambiente virtual de aprendizagem. Na Seção 6.1 é descrito o ambiente onde foram implementados os agentes neste estudo de caso. Na Seção 6.2 é apresentado o estudo de caso propriamente dito.

6.1 Descrição do ambiente virtual de aprendizagem utilizado no estudo de caso

O e-Grupo é um ambiente para apoiar a aprendizagem colaborativa na web e em dispositivos móveis que foi proposto inicialmente em (MENDES NETO, 2000) e que tem sido constantemente melhorado no Laboratório de Engenharia de Software (LES) da Universidade Federal Rural do Semi-Árido (UFERSA). Ele oferece um conjunto de aplicações que torna possível a interação, através da web e de dispositivos móveis (celulares, PDAs, etc), entre os participantes tanto de cursos presenciais quanto a distância para realização de trabalhos em grupo na modalidade de Aprendizagem Colaborativa Apoiada por Computador. Seu objetivo é prover os mesmos benefícios do ensino presencial. As ferramentas disponíveis no e-Grupo têm uma interface específica para dispositivos móveis, permitindo uma modalidade de aprendi-

zagem que tem sido denominada de Aprendizagem Móvel (MOURA; CARVALHO, 2008). O ambiente oferece uma interface simples, padronizada e interativa, permitindo aos usuários iniciantes se sentirem à vontade no uso de suas ferramentas.

Estas ferramentas são classificadas em Mecanismos de Coordenação, Mecanismos de Cooperação e Mecanismos de Comunicação, de acordo com o propósito a qual foram destinadas. Elas podem ser acessadas a qualquer hora, pois estão sempre visíveis para os usuários na forma de barra de ferramentas, tornando possível o controle total do ambiente, como pode ser visto na Figura 6.1.



Figura 6.1: Exemplo da interface web do e-Grupo

Existem cinco atores no e-Grupo:

1. **Facilitador:** É o responsável, entre outras coisas, pela condução dos trabalhos dos grupos. É geralmente alguém com formação pedagógica, capaz de auxiliar os alunos de EaD baseado na Web no processo de ensino-aprendizagem.
2. **Membro do Grupo:** É um aluno de EaD baseado na Web que está, em determinado momento, participando de um grupo para realização de um trabalho proposto pelo(s) Facilitador(es).
3. **Líder:** É um Membro do Grupo que difere dos demais por possuir algumas atribuições a mais, que lhe foram conferidas pelo(s) Facilitador(es), com a finalidade de aliviar a sobrecarga de trabalho deste(s).

Capítulo 6. Estudo de Caso: Detecção de Estudantes Passivos e Fuga de Contexto na PBL73

4. Administrador: É um superusuário com permissão total para executar as funções de administração do ambiente.
5. Usuário: É qualquer pessoa cadastrada no ambiente, independentemente de ser ou não Membro do Grupo, Líder, Facilitador ou Administrador.

As interfaces projetadas para os atores são semelhantes, diferenciando-se apenas pelas opções de Barras de Ferramentas. Por exemplo, os Facilitadores visualizam as três Barras de Ferramentas (Coordenação, Cooperação e Comunicação), enquanto os Membros do Grupo visualizam apenas as Barras de Ferramentas de Cooperação e Comunicação.

As ferramentas visualizadas em determinada Barra de Ferramentas também variam, dependendo das permissões definidas para os atores do e-Grupo. Inicialmente, algumas ferramentas são disponibilizadas para cada ator; no entanto estas permissões podem ser alteradas através da ferramenta de coordenação (Gerente de Permissão).

A Barra de Ferramentas de Coordenação apresenta as ferramentas responsáveis pela coordenação do ambiente (proporcionando os recursos necessários para gerenciar e configurar os diversos mecanismos de cooperação e comunicação oferecidos pelo ambiente. Ela também oferece recursos de interesse específico do administrador do ambiente, como, por exemplo, recursos de segurança. As ferramentas disponíveis na barra de ferramentas de coordenação também são responsáveis pela coordenação dos grupos, proporcionando os recursos necessários para os Facilitadores e Líderes controlarem eficazmente as atividades dos grupos.

A Barra de Ferramentas de Cooperação apresenta as ferramentas necessárias para a cooperação entre os participantes, possibilitando que os Membros do Grupo trabalhem remotamente em uma tarefa, eliminando a distância entre eles.

A Barra de Ferramentas de Comunicação apresenta as ferramentas utilizadas para a comunicação entre os participantes, tornando possível a realização de diversas modalidades de interação entre eles. Ela permite a troca de conhecimentos, experiências e expectativas em relação ao trabalho que está sendo desenvolvido (MENDES NETO, 2007; MENDES NETO; BRASILEIRO, 2005).

Uma descrição mais detalhada do e-Grupo foi feita no Apêndice A.

6.2 Estudo de Caso

Agentes inteligentes podem executar muitas tarefas na aprendizagem colaborativa apoiada por computador, tais como monitorar a participação dos estudantes em discussões, facilitar a seleção de tópicos para discussão, avaliar a performance do estudante em relação ao uso de ferramentas de comunicação

e cooperação disponíveis no ambiente, entre outras. O uso de agentes para auxiliar estas tarefas está ficando cada vez mais importante, principalmente devido ao crescimento do número de estudantes que interagem em sistemas de apoio ao aprendizado, o que torna muito difícil para os facilitadores gerenciar essas atividades a distância.

A abordagem baseada em agentes para detecção de estudantes passivos e de conversações fora do contexto do problema, proposta neste estudo de caso, está esboçada na Figura 6.2.

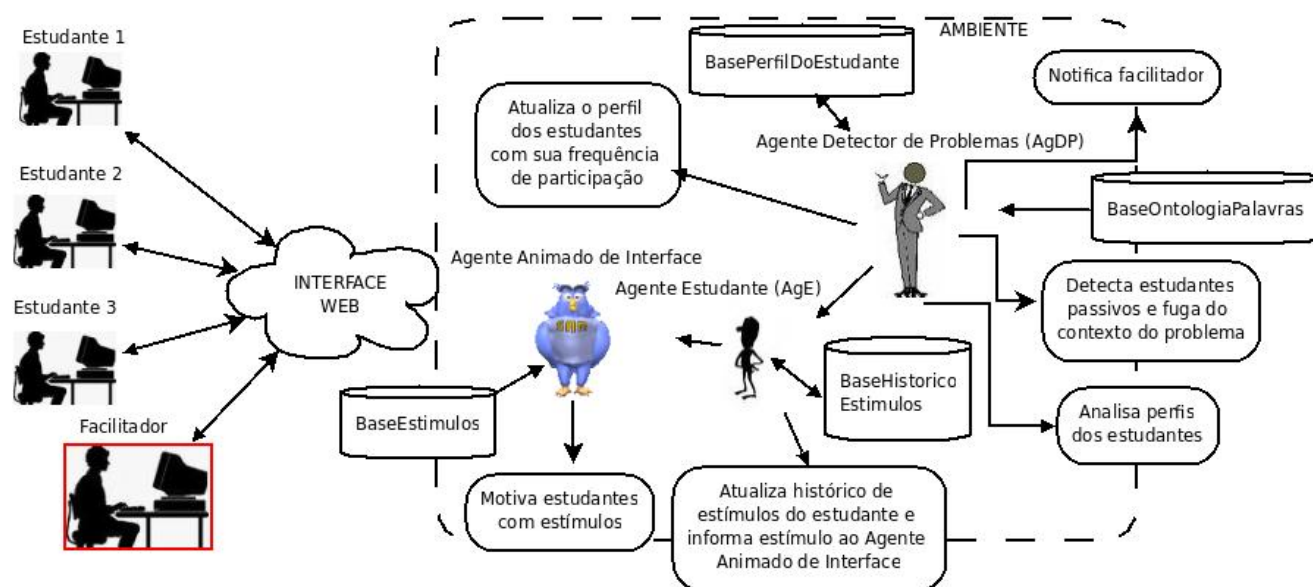


Figura 6.2: Abordagem para detecção de estudantes passivos e fuga do contexto do problema

De acordo com a abordagem apresentada na Figura 6.2, três tipos de agentes são propostos: um Agente Detector de Problemas (AgDP), um Agente Estudante (AgE) e um agente animado de interface. O AgDP é responsável por atualizar o perfil dos estudantes com a frequência de sua participação e sua frequência de conversações fora de contexto do problema sempre que os estudantes usam os mecanismos colaborativos. Estas informações serão usadas para detecção de estudantes passivos e fuga do contexto do problema, respectivamente, analisando o perfil dos estudantes existentes. Após detectar um estudante passivo ou a fuga do contexto, ele notifica ao AgE, que consulta em sua base de históricos se este estudante foi estimulado antes.

Se ele não tiver sido previamente estimulado, o AgE irá acionar o agente animado de interface que irá buscar pelo primeiro estímulo em sua base de estímulos que foram previamente cadastrados. Então o agente animado de interface irá tentar motivar o estudante com este estímulo. Em seguida, o AgE irá gravar o tipo de estímulo usado na sua base de histórico e notificar o facilitador sobre a situação do estudante.

Contudo, se o AgE detectar em sua base de históricos que o estudante foi previamente estimulado,

ele irá checar se já passou o prazo (previamente cadastrado pelo facilitador) que é dado para o estudante melhorar sua colaboração no ambiente de aprendizagem (ex. 15 dias). Se o prazo não passou, ele não faz nada. Mas se o prazo passou, ele notificará o agente animado de interface, que irá buscar o próximo estímulo em sua base de estímulos e notificará o facilitador sobre a situação do estudante.

Os agentes irão repetir este processo enquanto houver estímulos cadastrados na base de estímulos. O agente animado de interface é responsável por motivar os estudantes a participarem mais das discussões e a usarem as ferramentas disponíveis no ambiente virtual de aprendizagem. Estes agentes serão descritos em mais detalhes nas subseções seguintes.

6.2.1 Agente Detector de Problemas

O Agente Detector de Problemas (AgDP) é um agente reativo com estado interno. Seu estado interno reflete os perfis dos estudantes com o passar do tempo e percebe as ações executadas pelos estudantes. As funções *Ver* e *Próximo* do AgDP servem para atualizar os perfis dos estudantes com os dados sobre o uso das ferramentas colaborativas disponíveis no ambiente. Nesta abordagem, para cada ação executada pelo estudante no ambiente, o estado interno é alterado de acordo com uma abordagem baseada em pontuações, cujos valores são previamente definidos. Apenas para facilitar a compreensão deste estudo de caso, definimos as pontuações apresentados na Tabela 6.1.

Tabela 6.1: Tabela de pontuação de participação

Interação	Pontuação
Mensagem enviada pela sala de conversação	5
E-mail enviado	20
Mensagem na lista de discussão	10
Criação de grupo de interesse	30
Mensagem no grupo de interesse	10

Assim, para cada tipo de interação que os estudantes realizam durante a PBL, o AgDP atualiza os perfis dos estudantes descritos em seu estado interno de acordo com as pontuações definidas na Tabela 6.1. Esta informação será usada mais tarde pela função *Ação* do AgDP para detectar estudantes passivos.

A função *Ação* do AgDP é responsável por detectar estudantes passivos no processo da aprendizagem baseada em problemas. Os passos do processo de detecção de estudantes passivos executados pela função são descritos no Algoritmo 1.

Algoritmo 1 Ação: Detecção de estudantes passivos

Considerando :

estudante : *estudante resolvendo um mesmo problema na PBL*

grupo_estudantes : *conjunto de estudantes*

serie : *conjunto de pontos de um grupo de estudantes*

e : *elemento da série*

c1, c2 : *elementos centrais da série*

ordenar_serie(serie) : *ordena valores em ordem crescente*

quantidade_de_valores : *número de elementos da série*

lim_inf, lim_sup : *limites inferior e superior para valores discrepantes*

limiar_deteccao_passivos : *limite de participação definido*

lista_passivos : *armazena resultado do algoritmo*

```
1: para todo grupo_estudantes faça
2:   ordenar_serie(serie)
3:   se mod(quantidade_de_valores/2) = 0 então
4:     mediana = (c1 + c2)/2
5:   senão
6:     posicao_mediana = (quantidade_de_valores + 1)/2
7:     mediana = serie(posicao_mediana)
8:   fim se
9:   para todo e ∈ serie faça
10:    se e/mediana > lim_inf ∧ e/mediana < lim_sup então
11:      somatorio = somatorio + e
12:      numero_elementos = numero_elementos + 1
13:    fim se
14:  fim para
15:  media = somatorio/numero_elementos
16:  para todo estudante ∈ grupo_estudantes faça
17:    se e/media < limiar_deteccao_estudantes então
18:      lista_passivos = estudante
19:    fim se
20:  fim para
21:  retorne lista_passivos
22: fim para
```

Como pode ser visto no Algoritmo 1, inicialmente é calculada a média de participação dos estudantes, eliminando os valores discrepantes (*outliers*). A detecção de valores discrepantes é realizada calculando-

Capítulo 6. Estudo de Caso: Detecção de Estudantes Passivos e Fuga de Contexto na PBL77

se a mediana dos valores da série. A discrepância é eliminada de acordo com um limiar pré-estabelecido em relação a mediana dos valores. Em seguida, é calculada a média aritmética dos valores restantes, refletindo melhor a tendência da série. Com base na média dos valores restantes, é possível detectar um estudante passivo que diste do limiar pré-estabelecido pelo facilitador. Considerando, por exemplo, que $limiar_deteccao_passivos = 0.15$, $serie = (130, 140, 20, 135)$, $lim_inf = 0.2$ e $lim_sup = 1.5$, o estudante com pontuação igual a 20 seria detectado como estudante passivo.

O AgDP é também responsável por detectar estudantes que fugiram ao contexto durante as interações na PBL. O Algoritmo 2 apresenta os passos para alcançar esta meta.

Algoritmo 2 Ação: Detecção de fuga do contexto

Considerando :

- > *estudante* : *estudante resolvendo um mesmo problema na PBL*
- > *ferramenta_interacao* : *algum mecanismo para colaboração síncrona ou assíncrona*
- > *mensagem* : *alguma mensagem enviada por um estudante através de alguma ferramenta de interação*
- > *palavra* : *uma palavra em uma mensagem*
- > *OntologiaPalavrasRelacionadas* : *ontologia em que as palavras relacionadas ao contexto são instanciadas*
- > *PR* : *quantidade de palavras relacionadas ao contexto*
- > *OntologiaPalavrasNaoRelacionadas* : *ontologia em que as palavras não relacionadas ao contexto são instanciadas*
- > *PNR* : *quantidade de palavras não relacionadas ao contexto*
- > Δ : *fator de balanceamento usado pelo facilitador*
- > *NF* : *nível de fuga do contexto (guarda o resultado do algoritmo)*

```
1: para todo estudante faça
2:   para todo mensagem  $\in$  ferramenta_interacao faça
3:     para todo palavra  $\in$  mensagem faça
4:       se palavra  $\in$  OntologiaPalavrasRelacionadas então
5:          $PR = PR + 1$ 
6:       senão
7:         se palavra  $\in$  OntologiaPalavrasNaoRelacionadas então
8:            $PNR = PNR + 1$ 
9:         fim se
10:      fim se
11:    fim para
12:  fim para
13:   $NF = 2 * PNR / (PR - \Delta)$ 
14:  retorne NF
15: fim para
```

Como pode ser visto no Algoritmo 2, o AgDP monitora constantemente as ferramentas usadas pelos estudantes para cooperação e comunicação durante a resolução do problema. Então ele compara as palavras usadas pelos estudantes nas suas interações com um conjunto de palavras previamente instanciadas em uma ontologia de palavras relacionadas ao contexto do problema que está sendo resolvido. A Figura 6.3 mostra uma parte do código desta ontologia.

```
<owl:Class rdf:ID="PalavrasRelacionadasAoContexto">
  <rdfs:subClassOf rdf:resource="#Palavras"/>
</owl:Class>
<PalavrasRelacionadasAoContexto rdf:ID="Seminário"/>
<PalavrasRelacionadasAoContexto rdf:ID="Estudar"/>
<PalavrasRelacionadasAoContexto rdf:ID="Prova"/>
<PalavrasRelacionadasAoContexto rdf:ID="Disciplina"/>
<PalavrasRelacionadasAoContexto rdf:ID="Aula"/>
```

Figura 6.3: Parte do código da ontologia em OWL instanciada com algumas palavras relacionadas ao contexto do problema

A seguir, o AgDP irá também comparar estas palavras com palavras que certamente estão fora do contexto do problema que foram instanciadas em outra ontologia. A Figura 6.4 mostra uma parte do código desta ontologia.

```
<owl:Class rdf:ID="PalavrasForaDoContexto">
  <rdfs:subClassOf rdf:resource="#Palavras"/>
</owl:Class>
<PalavrasForaDoContexto rdf:ID="Festa"/>
<PalavrasForaDoContexto rdf:ID="Churrasco"/>
<PalavrasForaDoContexto rdf:ID="Boate"/>
<PalavrasForaDoContexto rdf:ID="Cerveja"/>
```

Figura 6.4: Parte do código da ontologia em OWL instanciada com algumas palavras fora do contexto do problema

Em seguida, o AgDP calcula a porcentagem de palavras usadas para cada estudante nas ferramentas disponíveis no ambiente fora do contexto do problema. Isto é útil para identificar o nível de fuga do contexto do estudante aos assuntos relacionados ao problema em discussão. Este nível pode ser obtido pela expressão $NF = 2 * PNR / (PR - \Delta)$, onde NF = Nível de Fuga; PNR = quantidade de palavras não relacionadas; PR = quantidade de palavras relacionadas; e Δ é um fator que o facilitador pode gerenciar para aumentar ou diminuir o impacto de palavras não relacionadas.

6.2.2 Agente Estudante

O Agente Estudante (AgE) é um agente reativo com estado interno. Seu estado interno reflete a base de históricos de estímulos usados para motivar os estudantes passivos e os estudantes envolvidos em conversações fora do contexto do problema detectados pelo AgDP. Ao longo do tempo, o AgE percebe a eficácia da aplicação de cada estímulo. A integração entre as funções *Ver* e *Próximo* serve para atualizar a base de históricos de estímulos com os dados sobre o uso deste estímulo para motivar os estudantes. Nesta abordagem, para cada estímulo usado pelo agente animado de interface no ambiente de aprendizagem, o estado interno do AgE será modificado para refletir que o estímulo já foi usado e sua

data de utilização. Baseado nesta data, o AgE saberá se o prazo previamente definido pelo facilitador para o estudante melhorar sua colaboração finalizou ou não. Isto é importante porque o estudante somente deverá ser motivado novamente se este prazo tiver expirado.

A Figura 6.5 mostra uma parte do código fonte do AgE.

```
...
public class AgEstudante {
    public String AvalieEstudante(String matusuario){
        String estimulo="";
        //Buscar histórico de estímulos
        try {
            ConnectionDB dbconn = new ConnectionDB();
            Statement strSQL = dbconn.getConnection().createStatement();
            String SQL="SELECT * FROM historico_estimulo where" +
                "matricula = '"+matusuario+"' order by id_estimulo desc";
            ResultSet rs = strSQL.executeQuery(SQL);
            if (rs.next()){
                //Encontrou entrada no histórico
                //Subtrair data atual da data do último estímulo
                Date ultimo_estimulo = rs.getDate("data");
                Calendar someDate = Calendar.getInstance();
                someDate.setTime( ultimo_estimulo );
                // Pega data atual no Calendário
                Calendar now = Calendar.getInstance();
                //Subtrai data atual da data do último estímulo
                long milliseconds1 = now.getTimeInMillis();
                long milliseconds2 = someDate.getTimeInMillis();
                long diff = milliseconds1 - milliseconds2;
                long diffDays = diff / (24 * 60 * 60 * 1000);
                if (diffDays > 15){
                    //Se houver passado o prazo, recuperar o próximo estímulo
                    int proximo_estimulo = rs.getInt("id_estimulo") + 1;
                    SQL="SELECT * FROM estimulo where id_estimulo = "+proximo_estimulo+"";
                    Statement strSQL4 = dbconn.getConnection().createStatement();
                    ResultSet rs4 = strSQL4.executeQuery(SQL);
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Figura 6.5: Parte do código fonte do agente estudante em Java

6.2.3 Agente Animado de Interface

O Agente Animado de Interface foi implementado para comunicação com os estudantes. Este agente tem por objetivo motivar os estudantes a participarem mais das discussões e a usarem as ferramentas colaborativas. Reategui, Boff e Campbell (2007) mostram que personagens de interface podem ter um impacto positivo na interação dos estudantes em ambientes de aprendizagem. Este agente gera mais confiança porque ele é socio-afetivo, ou seja, ele é capaz de expressar emoções através de animações, gestos e representações conhecidas, enfatizando suas características sociais.

A Figura 6.6 mostra um exemplo da ação do agente animado de interface no ambiente de aprendizagem e-Grupo, onde um estudante com comportamento passivo foi detectado.



Figura 6.6: Exemplo do agente animado de interface no e-Grupo

A Figura 6.7 mostra uma parte do código fonte do agente animado de interface.

```
if(session.getValue("MatUsuario")!=null){
//Verifica se o estudante é passivo
ArrayList<String> passivo = AgDP.DetectaEstudantesPassivos();

    if (passivo.contains(session.getValue("MatUsuario"))){

        String estimulo = AgEstudante.AvalieEstudante((String)session.getValue("MatUsuario"));

        if (!estimulo.equals("aguarde")){

            out.println("<script>");
            out.println("AgentIntro('"+session.getValue("NomeUsuario")+";");
            out.println("</script>");

            out.println("<script>");
            out.println("Sam.Show();");
            out.println("Sam.Speak('Deixe-me consultar alguns arquivos no meu notebook.');"");
            out.println("Sam.Play('WriteReturn');"");
            out.println("Sam.Play('WriteReturn');"");
            out.println("Sam.Play('Think');"");
            out.println("Sam.Play('Acknowledge');"");

            out.println("Sam.Speak('"+session.getValue("NomeUsuario")+"; "+estimulo+"');"");
            out.println("Sam.Play('Pleased');"");
            out.println("Sam.Play('DoMagic2');"");
            out.println("Sam.Play('Pleased');"");

            out.println("Sam.Speak('Boa Sorte com seus estudos!');"");
            out.println("Sam.Play('Wave');"");
            out.println("Sam.Hide();"");
            out.println("</script>");
        }
    }
}
```

Figura 6.7: Parte do código fonte do agente animado de interface em Java

O agente animado possui uma base de estímulos que possibilita que ele ofereça o estímulo ao estudante. O AgE irá comunicar-se com o agente animado de interface solicitando que ele ofereça um estímulo ou aguarde, dependendo do prazo de expiração presente na base de estímulos do AgE. O AgDP se comunica com o AgE passando a lista de estudantes passivos e/ou que fugiram ao contexto que têm sido detectados. Se o estudante presente na lista está on-line, o AgE pode solicitar a intervenção do agente animado de interface, que irá tentar motivá-lo com um estímulo de sua base.

6.3 Conclusão

Neste capítulo foi descrito o estudo de caso com a implementação do agente detector de problemas, agente estudante e agente de interface para cumprimento de duas metas: detecção de estudantes passivos e detecção de fuga do contexto. A arquitetura foi projetada para aumentar a colaboração em ambientes virtuais de aprendizagem seguindo os requisitos da PBL, este estudo de caso foi importante pois ressaltou metas relacionadas a colaboração e interação entre os estudantes e o ambiente virtual de aprendizagem.

Capítulo 7

Considerações finais e Trabalhos Futuros

Nos últimos anos, a EaD tem crescido e apresentado bons resultados, embora ainda apresente grandes desafios. Um destes desafios é a aplicação eficaz de teorias de aprendizagem no processo de ensino-aprendizagem nos ambientes que oferecem suporte informatizado a esta atividade. Muitos ambientes virtuais de aprendizagem enfocam aspectos relacionados à sua funcionalidade, esquecendo a função pedagógica do ambiente, no que tange à aplicação de uma teoria de aprendizagem para o sucesso do processo de ensino e aprendizagem.

A PBL, como visto ao longo deste trabalho, tem apresentado bons resultados com relação ao processo de ensino e aprendizagem, tais como: desenvolvimento do pensamento crítico e criatividade do estudante; aumento de sua capacidade de resolução de problemas; aumento de sua motivação entre outros. Contudo, foi visto também que sua implantação é complexa, pois necessita de uma série de requisitos para seu sucesso, como, por exemplo: o desenvolvimento de todo o seu ciclo, que não é um processo trivial; a definição de um problema de natureza complexa e mal estruturada; a responsabilidade de auto-aprendizado do estudante; a necessidade de o facilitador conhecer as técnicas da PBL; entre outros. A sua implantação na EaD é uma tarefa ainda mais complicada, pois o facilitador não está presente fisicamente nas sessões de interação e nem sempre possui informações sobre como o processo está sendo realizado ou se há algum problema a ser solucionado.

Outro grande desafio dos ambientes de aprendizagem de EaD é o crescimento do número de usuários nesta modalidade de ensino e aprendizagem. O uso de agentes de software tem se destacado como auxílio neste problema, facilitando o gerenciamento do ambiente e, ainda mais importante, o cumprimento de uma teoria de aprendizagem eficaz para o sucesso do processo de ensino e aprendizagem no ambiente. O uso de ontologias em ambientes de aprendizagem também tem se tornado comum, pois elas representam conceitos e modelos inerentes ao ambiente no qual são usadas, podendo modelar o conhecimento do

domínio e também auxiliar no processo de aprendizagem (FONTES; MENDES NETO; PONTES, 2009).

Este trabalho descreveu a análise e o projeto de uma arquitetura de agentes para suporte à colaboração na aprendizagem baseada em problemas em ambientes virtuais de aprendizagem (PONTES; MENDES NETO; CAMPOS, 2009, 2010a, 2010b). A arquitetura proposta neste trabalho foi modelada usando uma metodologia que foi selecionada através de um estudo realizado com outras nove metodologias de modelagens de sistemas multiagente. O estudo comparou diversos parâmetros, tais como: completude da metodologia, documentação existente, consolidação da metodologia, adaptação ao problema de agentes no contexto de ambientes de aprendizagem, suporte de ferramentas de apoio ao processo da metodologia e, por fim, aplicabilidade da metodologia. Deste estudo concluiu-se que a metodologia mais apropriada seria a metodologia Prometheus.

A PBL é uma teoria de aprendizagem que enfatiza a colaboração e o trabalho em grupo para resolução de um problema. No entanto, um problema que ocorre com frequência no processo de aplicação da PBL é a presença de estudantes passivos que, normalmente, são estudantes que possuem dificuldades de trabalhar em equipe e de colaborar para solução do problema. No ensino presencial, o facilitador pode detectar facilmente a presença de estudantes com este perfil e tentar corrigir esta situação para melhorar o processo de aprendizagem. Na EaD, por sua vez, isto não é uma tarefa trivial. Isto se deve, principalmente, a questões relacionadas à distribuição geográfica dos estudantes e à ausência de informações sobre o grau de motivação dos mesmos. Desta forma, neste trabalho foi feita a implementação da meta Detecção de Estudantes Passivos (ver Figura 5.2) como parte do estudo de caso da arquitetura proposta. Esta meta tem por objetivo detectar e corrigir esta situação indesejada, melhorando o processo de aprendizagem.

Assim como a presença de estudantes passivos é um problema na PBL, pois pode prejudicar o processo de aprendizagem, a fuga do contexto do problema por parte dos estudantes também é uma situação indesejada, pois na PBL o problema possui um papel essencial para o cumprimento da teoria. Assim, também foi implementado no estudo de caso a meta de detecção de conversações fora de contexto do problema (Detecção de fuga do contexto, ver Figura 5.2), através do monitoramento das conversações dos estudantes nas ferramentas colaborativas. Esta meta tem por objetivo focar as discussões realizadas no ambiente de aprendizagem em tópicos relacionados ao problema sendo solucionado na PBL.

Como contribuição deste trabalho podemos destacar o desenvolvimento de uma arquitetura baseada em agentes que poderá apoiar qualquer ambiente de aprendizagem colaborativa, seguindo os requisitos de uma teoria comprovadamente eficaz, ou seja, a PBL. Outra contribuição foi a implementação de uma parte da arquitetura de agentes para detecção de estudantes passivos e fuga do contexto do problema no ambiente virtual de aprendizagem e-Grupo. A arquitetura proposta irá facilitar a tarefa de ensino e aprendizagem quando a PBL for utilizada como teoria de aprendizagem. Ela irá também oferecer mais subsídio para que o facilitador possa acompanhar o processo de aplicação da PBL, possibilitando uma intervenção mais eficaz quando problemas de aprendizagem forem detectados.

Como trabalhos futuros, pretende-se: (i) Fazer um estudo de caso com uma turma para verificar o impacto da detecção de estudantes passivos e fuga do contexto do problema no grau de motivação dos estudantes. Este estudo de caso também terá como objetivo obter um embasamento maior para melhorar a estratégia de motivação adotada pelo agente animado, uma vez que motivar um estudante a usar uma ambiente virtual de aprendizagem para solucionar um problema, sem que esta intervenção gere frustrações no estudante, é uma tarefa complexa até mesmo para um agente humano; (ii) Abordar a detecção de problemas de colaboração, ou seja, verificar através de agentes se os estudantes estão efetivamente trabalhando em equipe; (iii) Abordar a detecção de estagnação do processo, que é um problema que ocorre quando os estudantes não possuem conhecimentos suficientes para continuar o processo, ou seja, o grupo fica “perdido”; (iv) Abordar o monitoramento do cumprimento das etapas, ou seja, incluir agentes para auxílio no cumprimento das etapas da PBL (definição do cenário do problema, análise do problema, avaliação e resolução do problema); (v) Abordar o problema de criação de grupos, ou seja, uma maneira automatizada de criação de grupos baseada no perfil dos estudantes.

Apêndice A

Descrição detalhada do ambiente e-Grupo

A.1 Ferramentas de Coordenação

Gerente de Permissão

Autoriza ou restringe permissão para utilizar os mecanismos de comunicação, cooperação e coordenação. O Gerente de Permissão disponibiliza uma interface simples e intuitiva que possibilita selecionar qualquer mecanismo do e-Grupo e definir quais atores poderão utilizá-lo. É o Gerente de Permissão que faculta alterar as permissões iniciais do e-Grupo, dando total flexibilidade para configuração das permissões de utilização de cada recurso do sistema. O Gerente de Permissão apresenta também uma interface para consultar as permissões de cada mecanismo do e-Grupo, ou seja, que atores podem utilizá-lo.

Gerente de Cadastro

É o responsável pelo cadastro dos participantes e dos grupos no e-Grupo. Está subdividido em: Cadastro de Usuários, Cadastro de Administradores, Cadastro de Facilitadores e Cadastro de Grupos.

Gerente de Comunicação

Ferramenta que permite definir os mecanismos de comunicação e cooperação síncronas que serão utilizados pelo grupo. O Gerente de Comunicação é utilizado pelo Gerente de Cadastro de Grupos para dar ou restringir permissão a determinado grupo para uso de cada um dos mecanismos de comunicação e

cooperação síncronas, como videoconferência, audioconferência, compartilhamento de aplicações, entre outros. Este tipo de restrição é particularmente útil em grupos cujo trabalho não exige tanta interação e quando os recursos de rede são limitados, visto que alguns mecanismos síncronos consomem, geralmente, muitos recursos de rede, em decorrência da necessidade de transportar, além dos dados, voz e/ou vídeo.

Gerente de Agenda

Ferramenta que possibilita agendar eventos, como por exemplo reuniões virtuais, controlando esta atividade de modo que não seja permitido agendar eventos em horários já reservados. Ela também oferece a opção de disponibilizar avisos, de interesse de determinado grupo, para todos os membros deste grupo.

Gerente de Materiais

Possibilita compartilhar materiais de apoio e resultados parciais e finais do trabalho, que podem ser arquivos ou referências para arquivos (URLs), a serem armazenados no Repositório de Materiais (que será explicado posteriormente). Ao compartilhar um material para determinado grupo, ele ficará armazenado em um diretório no servidor, só podendo ser visualizado e recuperado pelos Membros do Grupo ao qual pertence. Informações, como título, resumo, autor, relação com outro material e um indicador que torne possível criar links para ele de outras páginas do e-Grupo, deverão ser anexadas a cada material para facilitar a recuperação deste. O Gerente de Materiais permite estabelecer relações entre materiais compartilhados, de modo que um Membro do Grupo ou Facilitador, ao recuperar um material, possa disponibilizar outros que complementem a informação do primeiro. Estes materiais ficam relacionados no Repositório de Materiais e permitem que os outros Membros do Grupo identifiquem que materiais possuem relações com outros.

Gerente de FAQ

Possibilita publicar uma lista contendo as perguntas mais frequentes, e suas respectivas respostas, relacionadas a um trabalho, visando evitar duplicação de esforços com respostas a perguntas repetidas. Cada FAQ fica associada a um trabalho, só podendo ser visualizada pelos Membros dos Grupos que estão executando aquele trabalho.

Gerente de Trabalhos

Ferramenta para cadastro onde serão informadas as principais características dos trabalhos (título, descrição, data de entrega, grupos, Facilitadores, entre outras). É através do Gerente de Trabalhos que o Administrador define quais os grupos que irão executar determinado trabalho e quais os Facilitadores deste trabalho.

Gerente de Grupos de Interesse

Possibilita identificar pontos-chave do trabalho e definir Grupos de Interesse, a fim de estimular discussões direcionadas sobre estes pontos, visando melhorar a qualidade do trabalho final. O Gerente de Grupos de Interesse permite a definição de Grupos de Interesse específicos para cada grupo, de modo que dois grupos desenvolvendo o mesmo trabalho poderão ter Grupos de Interesse distintos.

Gerente de Sala de Conversação

Possibilita o gerenciamento da base de dados de conversações. O Gerente de Sala de Conversação apresenta uma interface para excluir conversações de determinado grupo. Todas as conversações realizadas através da ferramenta Sala de Conversação (que será explicada posteriormente) ficam armazenadas em um banco de dados e só podem ser excluídas através do Gerente de Sala de Conversação.

Gerente de Assistente

Mecanismo que permite criar Assistentes (ajuda on-line) para utilização das diversas ferramentas disponibilizadas no e-Grupo, indicando a adequação destas ferramentas para as diversas atividades pedagógicas. Ele também permite criar links para os Assistentes dos outros mecanismos.

A.2 Ferramentas de Cooperação

Cooperação Síncrona

Cooperador Síncrono: A ferramenta Cooperador Síncrono é, na realidade, um pacote de ferramentas que permite a um Membro do Grupo cooperar em tempo real com os outros membros do seu grupo e com os Facilitadores para resolução de problemas que requerem mais interação. O Cooperador Síncrono é composto pelas seguintes ferramentas:

- **Whiteboard:** consiste em uma área para desenho onde tudo que for desenhado por um dos usuários será visualizado pelos outros participantes da conferência e vice-versa. Útil para discutir modelos, diagramas, gráficos, fluxogramas, desenhos e outras coisas que facilitem o desenvolvimento do trabalho, geralmente é usado em conjunto com a ferramenta para conversação (*chat*) disponibilizada pelo Cooperador Síncrono.
- **Compartilhamento de Aplicações:** Permite a um Membro do Grupo compartilhar qualquer aplicação que esteja executando no seu computador com outros Membros do Grupo ou com os Facilitadores. O Compartilhamento de Aplicações é útil, entre outras coisas, para disponibilizar aplicações necessárias ao desenvolvimento do trabalho, de forma que Membros do Grupo, distantes geograficamente, possam compartilhar aplicações entre si.
- **Transferência de Arquivos:** Possibilita que os Membros do Grupo troquem arquivos entre si ou com Facilitadores enquanto estiverem utilizando o Cooperador Síncrono. Isto permite enviar ou recuperar arquivos que facilitem a compreensão de um assunto relacionado com a tarefa que está sendo realizada cooperativamente; além de enviar ou recuperar aplicações a serem executadas localmente para alcançar a performance desejada.

Cooperação Assíncrona

- **Repositório de Materiais:** Ferramenta que permite acessar os materiais do grupo (que podem ser arquivos ou referências para arquivos - URLs) e as informações associadas a eles, compartilhados através do Gerente de Materiais, e disponibilizar críticas e/ou sugestões, de modo que estas fiquem associadas ao material, podendo ser recuperadas junto com ele. Os materiais compartilhados no Repositório de Materiais ficam classificados em Materiais de Apoio e Resultados Parciais e Finais do Trabalho.
- **Arquivo de Trabalhos:** Ferramenta de consulta à base de dados dos trabalhos, que possibilita acessar as informações sobre os trabalhos dos grupos, como título, descrição, data de entrega, grupos responsáveis pelo trabalho, Facilitadores etc., de forma que faculta maior intercâmbio de informações entre os grupos e ajude os alunos a identificarem interesses comuns.
- **FAQ:** Permite acessar as perguntas mais frequentes sobre determinado trabalho publicadas através do Gerente de FAQ, constituindo-se em um meio rápido e eficaz para esclarecer dúvidas sobre assuntos relacionados ao trabalho.
- **Agenda:** Possibilita consultar os eventos (descrição do evento, data, hora, mecanismo de comunicação ou cooperação utilizado no evento etc.) agendados para determinado grupo, bem como os avisos (texto do aviso, data, hora etc.) disponibilizados para o grupo.

A.3 Ferramentas de Comunicação

Comunicação Síncrona

- **Sala de Conversação:** Permite aos Membros do Grupo conversarem entre si e com os Facilitadores em modo texto, e apresenta uma interface que exibe, juntamente com o comentário, a data, a hora e o nome de quem inseriu o comentário. Para cada grupo existente no ambiente é criada, automaticamente, uma Sala de Conversação, onde só poderão participar de conversações os Membros do Grupo e os Facilitadores do trabalho deste grupo. A Sala de Conversação apresenta também uma interface para consultar conversações anteriores, só permitindo que o Membro do Grupo visualize as conversações pertencentes ao seu grupo.
- **Audioconferência e Videoconferência Pessoal:** A ferramenta para Audioconferência é a mesma utilizada para Videoconferência Pessoal, e apresenta uma interface que possibilita ao Membro do Grupo participar de uma conferência com os outros Membros do Grupo e com os Facilitadores, utilizando recursos de áudio e/ou vídeo. Para cada grupo é criada uma conferência, da qual só poderão participar os Membros do mesmo grupo e os Facilitadores do trabalho do grupo. Existem três principais possibilidades de utilização das ferramentas de comunicação síncrona. O primeiro cenário corresponde a debates sobre pontos críticos do trabalho que necessitem de maior interação para se chegar a um senso comum. Para esta atividade podem ser utilizadas as seguintes ferramentas de comunicação: Sala de Conversação, Audioconferência e Videoconferência Pessoal. Outro possível cenário corresponde a apresentações de seminários em tempo real sobre pontos específicos do trabalho. Para esta atividade podem ser utilizadas as seguintes ferramentas de comunicação: Audioconferência e Videoconferência Pessoal. O último cenário corresponde a demonstrações de experimentos relacionados com o trabalho. Para esta atividade pode ser utilizada a Videoconferência Pessoal.

Comunicação Assíncrona

- **Correio Eletrônico:** Ferramenta que permite enviar mensagens relacionadas com o trabalho para determinado Membro do Grupo, para todos os Membros do Grupo, para os Facilitadores etc., apenas selecionando o(s) destinatário(s) de uma lista.
- **Lista de Discussão:** Ferramenta que permite aos Membros do Grupo discutirem entre si e com os Facilitadores sobre assuntos diversos, como por exemplo questões relacionadas ao trabalho. Ela também permite recuperar discussões anteriores.
- **Grupo de Interesse:** Ferramenta que permite discutir sobre pontos-chave do trabalho, identificados pelos Líderes e/ou Facilitadores, e recuperar discussões anteriores.

Outras funcionalidades

Para cada participante do e-Grupo, é disponibilizada uma Agenda Pessoal, dividida em quatro seções: Documentos, Anotações, Tarefas e Horário.

Na seção Documentos, o usuário pode armazenar em um diretório do Servidor qualquer arquivo que desejar. Isto facilita o gerenciamento de seus arquivos pessoais e torna possível acessá-los de qualquer lugar (de casa, do trabalho etc.). Esta seção também disponibiliza a data do arquivamento e permite alterar e excluir um documento a qualquer momento.

A seção Anotações possibilita ao usuário registrar qualquer anotação do seu interesse, que ficará, juntamente com a data, sempre visível. O usuário pode também alterar e excluir uma anotação quando necessário.

Na seção Tarefas, o usuário pode incluir as tarefas previstas para serem realizadas e seus prazos para conclusão. Inicialmente a tarefa aparece com o status pendente, que será alterado para concluído quando finalizada. Esta seção possibilita ainda alterar os dados de uma tarefa (descrição, prazo para conclusão e status) e excluí-la.

A seção Horário consiste em uma matriz dos dias da semana e dos turnos (Manhã, Tarde e Noite), onde o usuário poderá organizar seu horário semanal, alterando-o à vontade.

A.4 Implementação do e-Grupo

O e-Grupo é um ambiente distribuído, que pode ser acessado de qualquer lugar através de um navegador (cliente universal) ou dispositivos móveis. Ele foi implementado numa linguagem orientada a objetos, que torna a inclusão de novos componentes relativamente simples. Devido ao fato de ter sido implementado na linguagem Java, o e-Grupo é independente da plataforma do servidor. Pelo fato de todo o processamento acontecer no servidor, usando *Servlets*¹, ele também é independente da plataforma do cliente. O e-Grupo pode usar bancos de dados construídos em diferentes plataformas, pois ele usa a API JDBC (*Java Database Connectivity*) para conexão com o banco de dados.

O e-Grupo foi estruturado usando a arquitetura em três camadas (*3-tier architecture*), onde a interface do cliente - navegador ou interface móvel (primeira camada) - é separada da aplicação - Servidor web com *Servlets* (camada do meio), que é separada dos dados - servidor de banco de dados

¹Servlets são programas Java carregados e executados por um servidor que estendem a funcionalidade do servidor. Eles permitem aos programadores estenderem a funcionalidade de qualquer servidor que suporte Java, como um servidor web, um servidor de correio eletrônico, um servidor de aplicação, entre outros. Servlets fornecem alto grau de portabilidade, flexibilidade e facilidade (MENDES NETO, 2000).

(terceira camada).

As classes implementadas no e-Grupo estão contidas em quatro pacotes:

1. MecanismosCoordenacao, que contém dezessete classes referentes às ferramentas de coordenação do ambiente;
2. MecanismosCooperacao, que contém quatro classes das ferramentas de cooperação;
3. MecanismosComunicacao, composto por três classes das ferramentas de comunicação; e
4. AgendaPessoal, que contém três classes que controlam a agenda pessoal do usuário.

Algumas das funcionalidades das ferramentas do e-Grupo foram implementadas em JSP (*Java Server Pages*)².

A.5 Cenário de Uso do e-Grupo

Um típico uso do e-Grupo acontece como visto a seguir. Inicialmente o usuário fornece seu login e sua senha; o Sistema valida a senha; ele verifica o status do Usuário e o redireciona para uma interface específica para aquele ator. Se o ator for um Facilitador, o Sistema apresenta uma lista contendo todos os trabalhos orientados por ele. Ele seleciona, então, um trabalho e o sistema disponibiliza uma lista contendo todos os Grupos que estão executando aquele trabalho. Caso deseje cooperar ou se comunicar com algum Grupo, o Facilitar escolhe entrar em um Grupo e está apto para utilizar qualquer mecanismo de comunicação e/ou cooperação. Ele pode interagir, de forma síncrona ou assíncrona, com todos os membros daquele Grupo ou com qualquer Membro do Grupo individualmente.

Por exemplo, para participar de uma Videoconferência com os Membros do Grupo, o Facilitador seleciona o ícone da Videoconferência na Barra de Ferramentas de Comunicação. Esta barra de ferramentas, juntamente com as Barras de Ferramentas de Cooperação e de Coordenação, estão sempre visíveis para ele. Então a aplicação de videoconferência é carregada, já conectada na conferência criada para aquele Grupo. Um vez conectada em uma videoconferência específica, ele pode interagir com os Membros do Grupo e os Facilitadores do trabalho do Grupo. Ao entrar em um determinado Grupo, o Facilitador pode também realizar atividades de coordenação, que só refletem naquele Grupo, como, por exemplo, disponibilizar materiais de apoio ou agendar eventos para o Grupo.

O cenário de uso de outros atores é similar ao do facilitador, discutido acima, obviamente, limitado às funcionalidades fornecidas pelas barras de ferramentas de cada tipo de usuário.

²A grande vantagem desta abordagem é a possibilidade de separação entre a lógica da aplicação (código Java) e a apresentação (HTML). Esta separação é muito útil para o programador. Outra vantagem é que JSP não requer mudanças na API Servlet (HUNTER; CRAWFORD, 1998).

Referências bibliográficas

AUSUBEL, D. *The psychology of meaningful verbal learning: An introduction to school learning*. [S.l.]: Grune & Stratton New York, 1963.

AYALA, G.; YANO, Y. Evaluating the performance of agents that support the effective collaboration of learners in a CSCL environment. *IEICE Transactions on Information and Systems*, n. 2, p. 125–134, 1997.

AZEVEDO H. ; SCALABRIN, E. E. A human collaborative learning environment using intelligent agents. In: LIN, F. O. (Ed.). *Designing distributed Learning Environments with Intelligent Software Agents*. [S.l.]: Information Science Publishing, 2005. p. 1–32.

AZEVEDO, H. D. *Contribution à la modélisation des connaissances à l'aide des systèmes multi-agents*. Tese (Doutorado) — Université de Technologie de Compiègne, 1997.

BASTOS FILHO, O. Sistema Inteligente de Desafios Abertos–IOCS: uma proposta de adaptação dos padrões do Método Clínico Piagetiano utilizando provas lógicas de concepções abertas em Sistema Multiagentes. *Informática na educação: teoria & prática*, 2006.

BELLIFEMINE, F.; CAIRE, G.; GREENWOOD, D. *Developing multi-agent systems with JADE*. [S.l.]: Wiley, 2007.

BERNON, C. et al. Tools for Self-Organizing Applications Engineering. *Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering*, Springer, p. 283, 2004.

BERNON, C. et al. Engineering adaptive multi-agent systems: The adelfe methodology. In: HENDERSON-SELLERS; GIORGINI, P. (Ed.). *Agented-Oriented Methodologies*. [S.l.]: IDEA Group Publishing, 2005. cap. VII, p. 172–202.

BERNON, C. et al. ADELFE: A Methodology for Adaptive Multi-Agent Systems Engineering. Petta, P., Tolksdorf, R., and Zambonelli, F. In: *Proceedings of the rd International Workshop on Engineering Societies in the Agents World (ESAW 2002)*. [S.l.: s.n.], 2002. p. 156–169.

BITTENCOURT, I. et al. Ontologia para Construção de Ambientes Interativos de Aprendizagem. In: *XVII Simposio Brasileiro de Informatica na Educaçao, SBIE, Brasília*. [S.l.: s.n.], 2006. p. 559–568.

- BRESCIANI, P. et al. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, Springer, v. 8, n. 3, p. 203–236, 2004.
- BURRAFATO, P.; COSSENTINO, M. Designing a multi-agent solution for a bookstore with the PASSI methodology. In: *Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002)*. [S.l.: s.n.], 2002. p. 27–28.
- BUSETTA, P. et al. Jack intelligent agents-components for intelligent agents in java. *AgentLink News Letter*, v. 2, p. 2–5, 1999.
- CAO, Y.; GREER, J. Facilitating web-based education using intelligent agent technologies. *Proceedings of the Second International Workshop on Web-based Support System (WSS 2004)*, p. 37–44, 2004.
- CARLISLE, C.; IBBOTSON, T. Introducing problem-based learning into research methods teaching: Student and facilitator evaluation. *Nurse Education Today*, Elsevier, v. 25, n. 7, p. 527–541, 2005.
- CASTAÑON, G. Construtivismo e Ciências Humanas. *Ciências & Cognição*, v. 5, p. 36–49, 2005.
- CASTAÑON, G. Construtivismo, Inatismo e Realismo: compatíveis e complementares. *Ciências & Cognição*, v. 10, p. 115–131, 2007.
- CHEN, W.; WASSON, B. Intelligent agents supporting distributed collaborative learning. In: LIN, F. O. (Ed.). *Designing distributed learning environments with intelligent software agents*. [S.l.]: Information Science Publishing, 2005. p. 33–66.
- COSSENTINO, M. From requirements to code with the passi methodology. In: HENDERSON-SELLERS; GIORGINI, P. (Ed.). *Agented-Oriented Methodologies*. [S.l.]: IDEA Group Publishing, 2005. cap. IV, p. 79–106.
- COSSENTINO, M.; LOMBARDO, S.; SABATUCCI, L. *PASSI Toolkit (PTK)*. 2010. Acessado em 30 de Abril de 2010. Disponível em: <<http://sourceforge.net/projects/ptk>>.
- COUTINHO, C.; JUNIOR, B.; BATISTA, J. Collaborative learning using Wiki: a pilot study with master students in educational technology In Portugal. In: *Proceedings of World Conference on Educational Multimedia, Hypermedia, and Telecommunications*. [S.l.: s.n.], 2007. p. 1786–1791.
- DELOACH, S.; ENGINEERING, A. F. I. O. T. W.-P. A. O. S. O.; MANAGEMENT. *Analysis and Design using MaSE and agentTool*. [S.l.]: AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING AND MANAGEMENT, 2001.
- DELOACH, S.; KUMAR, M. Multi-agent systems engineering: An overview and case study. In: HENDERSON-SELLERS; GIORGINI, P. (Ed.). *Agented-Oriented Methodologies*. [S.l.]: IDEA Group Publishing, 2005. cap. XI, p. 317–340.

DILLENBOURG, P. What do you mean by collaborative learning. *Collaborative learning: Cognitive and computational approaches*, p. 1–16, 1999.

DIMITRACOPOULOU, A. Designing collaborative learning systems: current trends & future research agenda. In: INTERNATIONAL SOCIETY OF THE LEARNING SCIENCES. *Proceedings of th 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!* [S.l.], 2005. p. 115–124.

DING, Y.; WANG, Q.; HUANG, J. The performance optimization of clips. In: *Hybrid Intelligent Systems, 2009. HIS '09. Ninth International Conference on*. [S.l.: s.n.], 2009. p. 417 –421.

FALCÃO, J. A. *Modelagem Conceitual da Memória Organizacional Integrando processos de Negócios com a Engenharia do Conhecimento*. Dissertação (Mestrado) — Universidade de Fortaleza, 2002.

FONTES, L. M. O.; MENDES NETO, F. M.; PONTES, A. A. A. Uma ontologia sobre aprendizagem colaborativa apoiada por computador. In: *Escola Potiguar de Computação e suas Aplicações - EPOCA 2009, Natal. Anais da Escola Potiguar de Computação e suas Aplicações*. [S.l.]: EDUFRN Editora da UFRN, 2009.

FUENTES, R.; GOMEZ-SANZ, J.; PAVON, J. Activity theory for the analysis and design of multi-agent systems. *Lecture Notes in Computer Science 2935*, Springer, p. 110–122, 2003.

GARIJO, F. et al. The message methodology for agent-oriented analysis and design. In: HENDERSON-SELLERS; GIORGINI, P. (Ed.). *Agented-Oriented Methodologies*. [S.l.]: IDEA Group Publishing, 2005. cap. VIII, p. 203–234.

GASCUENA, J.; FERNANDEZ-CABALLERO, A.; GONZALEZ, P. Domain ontology for personalized e-learning in educational systems. *Advanced Learning Technologies, 2006. Sixth International Conference on*, p. 456–458, July 2006.

GLADUN, A. et al. An application of intelligent techniques and semantic web technologies in e-learning environments. *Expert Systems With Applications*, Elsevier, v. 36, n. 2P1, p. 1922–1931, 2009.

GOMES, P. et al. Using Ontologies for eLearning Personalization. In: *3rd ELearning Conference- Computer Science Education, Coimbra, Portugal*. [S.l.: s.n.], 2006.

GONÇALVES, M. d. L. S. *Portfolio Aprendizagem*. 2010. Acessado em 30 de Abril de 2010. Disponível em: <<http://portfolio.alfarod.net/>>.

GONZÁLEZ, L. A. G. *Um modelo conceitual para aprendizagem colaborativa baseada na execução de projetos pela Web*. Tese (Doutorado) — Universidade de São Paulo, Escola Politécnica, 2005.

GREENE, R.; ROGERS, G. Justifying core faculty assessment of students clinical performance using cognitive flexibility theory: A case example. *The Internet Journal of Allied Health Sciences and Practice*, v. 4, n. 3, 2006.

HENDERSON-SELLERS, B.; GIORGINI, P. *Agent-oriented Methodologies*. [S.l.]: Idea Group Inc (IGI), 2005. ISBN 1591405815.

HMELO-SILVER, C. Problem-Based Learning: What and How Do Students Learn? *Educational Psychology Review*, v. 16, n. 3, 2004.

HMELO-SILVER, C.; BARROWS, H. Goals and strategies of a problem-based learning facilitator. *The Interdisciplinary Journal of Problem-based Learning*, v. 1, n. 1, p. 21–39, 2006. Disponível em: <<http://docs.lib.purdue.edu/ijpbl/vol1/iss1/4>>.

HUNTER, J.; CRAWFORD, W. *Java Servlet Programming*. Sebastopol, CA: O'Reilly & Associates, 1998. 511 p.

HUTCHINS, E. *Cognition in the Wild*. [S.l.]: Cambridge, MA: MIT Press, 1995.

IGLESIAS, C. *Definición de una Metodología para el Desarrollo de Sistemas Multiagente*. Tese (Doutorado) — Departamento de Ingeniería de Sistemas Telemáticos, Universidade Politécnica de Madri, Madri, 1998.

IGLESIAS C.A. E GARIJO, M. The agent oriented methodology mascommonkads. In: HENDERSON-SELLERS; GIORGINI, P. (Ed.). *Agented-Oriented Methodologies*. [S.l.]: IDEA Group Publishing, 2005. cap. III, p. 46–78.

ISRAEL, J.; AIKEN, R. Supporting Collaborative Learning With An Intelligent Web-Based System. *International Journal of Artificial Intelligence in Education*, IOS Press, v. 17, n. 1, p. 3–40, 2007.

LAI, Y. Teaching computer applications to pre-school teachers through problem based learning approach. *ACM SIGCSE Bulletin*, ACM New York, NY, USA, v. 37, n. 4, p. 89–92, 2005.

LANGLEY, P.; LAIRD, J. E.; ROGERS, S. Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, v. 10, n. 2, p. 141–160, 2009.

LIU, X. et al. I-MINDS: an application of multiagent system intelligence to on-line education. In: *IEEE International Conference on Systems, Man and Cybernetics, 2003*. [S.l.: s.n.], 2003. v. 5.

LUCENA, C. J. P. et al. Communication, coordination and cooperation in computer-supported learning: The aulanet experience. In: NETO, F. M. M.; BRASILEIRO, F. V. (Ed.). *Advances in Computer-Supported Learning*. Hershey, PA, USA: Information Science Publishing, 2006.

MAES, P. Artificial life meets entertainment: lifelike autonomous agents. *Communications of the ACM*, ACM Press New York, NY, USA, v. 38, n. 11, p. 108–114, 1995.

MARQUES, J. Aprendizagem colaborativa: atividades de grupo como núcleo e uso do computador como contexto. *Psicol. argum*, v. 24, n. 44, p. 37–43, 2006.

MENDES NETO, F. *E-grupo: Um ambiente para suporte à aprendizagem colaborativa baseada na web*. Dissertação (Mestrado) — Universidade Federal de Campina Grande, Campina Grande, PB, 2000.

MENDES NETO, F.; BRASILEIRO, F. e-Grupo: Um ambiente para Suporte à Aprendizagem Colaborativa Baseada na Web. In: *Congresso da Sociedade Brasileira de Computação SBC, 21, Workshop de Informática na Escola WIE*. [S.l.: s.n.], 2001. v. 7, p. 610–621.

MENDES NETO, F.; BRASILEIRO, F. Uma Taxonomia para Ambientes de Aprendizagem Suportados pela Web. In: *Congresso da Sociedade Brasileira de Computação, 22, SBC 2002, Workshop de Informática na Escola WIE*. [S.l.: s.n.], 2002. v. 8, p. 957–966.

MENDES NETO, F. M. Analysis of the contribution of computer-supported learning to organizational knowledge management. In: *EATIS 2007 - Euro American Conference on Telematics and Information Systems*. Faro, Portugal: ACM Proceedings, 2007.

MENDES NETO, F. M.; BRASILEIRO, F. V. Integrating computer-supported learning and knowledge management through the support of the training management life cycle. In: *III Conference Professional Knowledge Management, I Workshop on Learner-Oriented Knowledge Management & KM-Oriented E-Learning (LOKMOL 2005)*. Kaiserslautern, DE: Springer, 2005. p. 152–155.

MIN, W.; WEI, C.; LEI, C. Research of ontology-based adaptive learning system. *Computational Intelligence and Design, 2008. ISCID '08. International Symposium on*, v. 2, p. 366–370, October 2008.

MOISIL, I. et al. Socio-cultural modelling of the student as the main actor of a virtual learning environment. In: *Proc. of the 8th WSEAS Int. Conf. on Mathematical Methods and Computational Techniques in Electrical Engineering*. [S.l.: s.n.], 2006. v. 119.

MOURA, A.; CARVALHO, A. Mobile Learning: Teaching and Learning with Mobile Phones and Podcasts. In: *Proceedings of the 2008 Eighth IEEE International Conference on Advanced Learning Technologies*. [S.l.]: IEEE Computer Society, 2008. p. 631–633.

MUUKKONEN, H.; HAKKARAINEN, K.; LAKKALA, M. Collaborative technology for facilitating progressive inquiry: Future learning environment tools. In: INTERNATIONAL SOCIETY OF THE LEARNING SCIENCES. *Proceedings of the 1999 conference on Computer support for collaborative learning*. [S.l.], 1999.

MYLOPOULOS, J. C. P. G. S. K. J. Tropos: A requirements-driven methodology for agent-oriented software. In: HENDERSON-SELLERS; GIORGINI, P. (Ed.). *Agent-Oriented Methodologies*. [S.l.]: IDEA Group Publishing, 2005. cap. II, p. 20–45.

NIQUINI, D.; BOTELHO, F. Telemática na Educação. *Tecnologia Educacional*, v. 29, n. 146, p. 27–33, 1999.

ODELL, J.; PARUNAK, H.; BAUER, B. Extending UML for Agents. *Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligence*, p. 3–17, 2000.

O'KELLY, J.; GIBSON, J. Robocode & problem-based learning: a non-prescriptive approach to teaching programming. In: ACM NEW YORK, NY, USA. *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education*. [S.l.], 2006. p. 217–221.

PADGHAM, L.; THANGARAJAH, J.; WINIKOFF, M. Tool support for agent development using the Prometheus methodology. In: *Quality Software, 2005. (QSIC 2005). Fifth International Conference on*. [S.l.: s.n.], 2005. p. 383–388.

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

PADGHAM, L.; WINIKOFF, M. *Developing intelligent agent systems: a practical guide*. [S.l.]: John Wiley and Sons, 2004.

PADGHAM L. WINIKOFF, M. Prometheus:a practical agent-oriented methodology. In: HENDERSON-SELLERS; GIORGINI, P. (Ed.). *Agented-Oriented Methodologies*. [S.l.]: IDEA Group Publishing, 2005. cap. V, p. 107–135.

PAVÓN, J.; GÓMEZ-SANZ, J.; FUENTES, R. The ingenias methodology and tools. In: HENDERSON-SELLERS; GIORGINI, P. (Ed.). *Agented-Oriented Methodologies*. [S.l.]: IDEA Group Publishing, 2005. cap. IX, p. 236–276.

PIAGET, J. Psicologia e epistemologia: Por uma teoria do conhecimento (A. Cretella, Trad.). *São Paulo: Ed. Forense (original publicado em 1966)*, 1973.

PONTES, A. A. A.; MENDES NETO, F. M.; CAMPOS, G. A. L. d. Uma arquitetura de agentes para detecção de estudantes passivos na aprendizagem baseada em problemas em ambientes virtuais de aprendizagem. In: *II Workshop sobre Avaliação e Acompanhamento da Aprendizagem em Ambientes Virtuais, XX Simpósio Brasileiro de Informática na Educação, SBIE 2009, Florianópolis. Anais do XX Simpósio Brasileiro de Informática na Educação*. [S.l.]: UFSC, 2009.

PONTES, A. A. A.; MENDES NETO, F. M.; CAMPOS, G. A. L. d. Multiagent system for detecting passive students in problem-based learning. In: *8th International Conference on Practical Applications of Agents and Multi-Agent Systems - PAAMS 2010, Salamanca, Spain. Proceedings of the 8th International Conference on Practical Applications of Agents and Multi-Agent Systems*. [S.l.]: Springer Verlag, 2010.

PONTES, A. A. A.; MENDES NETO, F. M.; CAMPOS, G. A. L. d. Multiagent system for detecting passive students in problem-based learning. *Advances in Soft Computing*, v. 71, p. 165–172, 2010.

PUNTAMBEKAR, S. An Integrated Approach to Individual and Collaborative Learning in a Web-Based Learning Environment. In: *Computer Supported Collaborative Learning, CSCL '99*. Stanford, CA: Stanford University, 1999. p. 458–465.

REATEGUI, E.; BOFF, E.; CAMPBELL, J. A. Using virtual characters in personalized recommendations. In: *SIGCSE '07: Proceedings of the 38th SIGCSE technical symposium on Computer science education*. New York, NY, USA: ACM, 2007. p. 180–184. ISBN 1-59593-361-1.

RIBAS, A.; MOURA, M. de. Abordagem sociocultural: algumas vertentes e autores. *Psicol. estud*, SciELO Brasil, v. 11, n. 1, 2006.

RISSOLI, V.; GIRAFFA, L.; MARTINS, J. de P. Sistema Tutor Inteligente baseado na Teoria da Aprendizagem Significativa com acompanhamento Fuzzy. *Informática na educação: teoria & prática*, v. 9, n. 2, 2008.

RIZZI, C. B. *A cooperação na ação e uma especificação de requisitos para agentes e sistemas multiagente fundamentadas na epistemologia genética*. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul. Faculdade de Educação. Programa de Pós-Graduação em Informática na Educação., 2006. Disponível em: <<http://hdl.handle.net/10183/12866>>.

ROGERS, Y. A brief introduction to distributed cognition. <http://mcs.open.ac.uk/yr258/papers/dcog/dcog-brief-intro.pdf>, 1997.

ROGERS, Y. Distributed cognition and communication. *The Encyclopedia of Language and Linguistics*, p. 181–202, 2006.

RUMBAUGH, J. et al. *Object-oriented modeling and design*. [S.l.]: Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1991.

RUSSELL, R.; NORVIG, P. *Artificial Intelligence, a Modern Approach 2nd Edition*. [S.l.]: Prentice Hall, 2003.

SAVERY, J. Overview of problem-based learning: Definitions and distinctions. *The Interdisciplinary Journal of Problem-based Learning*, v. 1, n. 1, p. 9–20, 2006.

SCHREIBER, G. *Knowledge engineering and management: the CommonKADS methodology*. [S.l.]: MIT press, 2000.

ŞENDAĞ, S.; ODABAŞI, H. F. Effects of an online problem based learning course on content knowledge acquisition and critical thinking skills. *Computers & Education*, Elsevier, 2009.

- SOH, L.-K. et al. A computer-supported cooperative learning system with multiagent intelligence. In: *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2006. p. 1556–1563. ISBN 1-59593-303-4.
- SPIRO, R. et al. Cognitive Flexibility Theory: Hypermedia for Complex Learning, Adaptive Knowledge Application, and Experience Acceleration. *Educational Technology*, v. 43, n. 5, p. 5–10, 2003.
- STROBEL, J.; BARNEVELD, A. van. When is PBL More Effective? A Meta-synthesis of Meta-analyses Comparing PBL to Conventional Classrooms. *Interdisciplinary Journal of Problem-based Learning*, v. 3, n. 1, p. 4, 2009.
- SUEBNUKARN, S.; HADDAWY, P. Comet: A collaborative tutoring system for medical problem-based learning. *Intelligent Systems, IEEE*, v. 22, n. 4, p. 70–77, July-Aug. 2007. ISSN 1541-1672.
- TSENG, K.; CHIANG, F.; HSU, W. Interactive processes and learning attitudes in a web-based problem-based learning (PBL) platform. *Computers in Human Behavior*, Elsevier, v. 24, n. 3, p. 940–955, 2008.
- VIGOTSKI, L. *A formação social da mente: o desenvolvimento dos processos psicológicos superiores*. [S.l.]: 6ª ed. São Paulo: Martins Fontes, 1998.
- WEISS, G. *Multiagent systems: a modern approach to distributed artificial intelligence*. [S.l.]: The MIT press, 1999.
- WOOLDRIDGE, M. *An Introduction to Multiagent Systems*. [S.l.]: John Wiley & Sons, Inc. New York, NY, USA, 2002.
- YEO, J.; TAN, S.; LEE, Y. A learning journey in problem-based learning. In: INTERNATIONAL SOCIETY OF THE LEARNING SCIENCES. *Proceedings of the 7th international conference on Learning sciences*. [S.l.], 2006. p. 859–865.
- YU, E. *Modelling strategic relationships for process reengineering*. [S.l.]: University of Toronto Toronto, Ont., Canada, Canada, 1995.
- YUEH, H.; LIN, W. Developing a Web-based environment in supporting students team-working and learning in a problem-based learning approach. In: *Creating, Connecting and Collaborating through Computing, 2005. C5 2005. Third International Conference on*. [S.l.: s.n.], 2005. p. 145–149.
- ZAMBONELLI, F.; JENNINGS, N.; WOOLDRIDGE, M. Developing multiagent systems: The Gaia methodology. *ACM Transactions on software Engineering and Methodology*, v. 12, n. 3, p. 317–370, 2003.
- ZAMBONELLI, F.; JENNINGS, N.; WOOLDRIDGE, M. Multi-agent systems as computational organizations: The gaia methodology. In: HENDERSON-SELLERS; GIORGINI, P. (Ed.). *Agented-Oriented Methodologies*. [S.l.]: IDEA Group Publishing, 2005. cap. VI, p. 136–171.